APPENDIX A

PROVISIONAL PATENT APPLICATION

for

INTEGRATED ACCESS DEVICE FOR ASYNCHRONOUS TRANSFER MODE (ATM) COMMUNICATIONS

Inventors: Kenneth Wayne Brinkerhoff
27825 Terales
Mission Viejo, California 92692

Wayne Paul Boese
2053-A Tustin Avenue
Costa Mesa, California 92627

Robert C. Hutchins
24272 Solonica Street
Mission Viejo, California 92692

Stanley (NMI) Wong
2509 West Hall Avenue
Santa Ana, California 92704

Attorney for Applicant:

William G. Lane, Inc., P.C.
18400 Von Karman Avenue, Suite 500
Irvine, California 92612
Telephone: 949 474-9961
Telefax: 949 474-9973

Attorney Docket No. M015-1001 Prov

Express Mail No. EJ130608565US

Date Mailed: June 30, 2000

INTEGRATED ACCESS DEVICE FOR ASYNCHRONOUS TRANSFER MODE (ATM) COMMUNICATIONS

BACKGROUND OF THE INVENTION

A. Field of the Invention

The present invention relates to the communication of information by electrical or optical signals. More particularly, the invention relates to an integrated access device apparatus and method for accessing digital information signals transmitted in an Asynchronous Transfer Mode (ATM), and for converting voice, video and data information to ATM signals for transmission.

Description of Background Art

Enterprises such as private companies, learning institutions, health care organizations and governmental agencies routinely must transfer information in a substantially instantaneous or "real time" fashion between locations which are too far apart to permit face-to-face contact. Such information transfers include voice and telefacsimile transmissions over existing telephone communication channels, digital data interchange between computers, including Internet communications, and video conferencing.

Many enterprises also utilize a network of computer work stations located in individual offices or cubicles, which are interconnected with each other and sometimes with a larger computer which functions as a Server for the network. A Server typically has substantially greater memory storage and/or computational power than individual PCs (Personal Computers) located at employee work stations, and thus is often an expedient economic choice because the greater processing and memory capabilities of the Server, with the concomitant increases in size, power consumption and cost that these increased capabilities entail, need not be replicated in each work station PC.

A variety of network interconnection configurations, or topologies are employed in the interconnection of computers at a given enterprise site. Such networks are frequently referred to as Local Area Networks or LANs because of the relatively close geographic proximity of the interconnected computers. A popular interconnection standard and data exchange protocol for LANs is referred to as the Ethernet.

LANs as described above may be linked together to form a higher level, i.e., more broadly inclusive, network connecting geographically separated offices in a city, in a Metropolitan Area

Network (MAN). MANs can be linked together to form a Wide Area Network (WAN), which might stretch nationwide, or to a worldwide network or Global Area Network (GAN), such as the Internet.

Existing telephone communication lines which link telephones world-wide employ a hierarchical interconnection scheme similar to that used between LANs at the user-end, node or "Edge" at one end of a network, and the GAN spanning the globe at the other end. Thus, enterprise sites are frequently equipped with Private Branch Exchanges (PBXs) that interconnect telephones and enable telephone communications between employees at a particular site. Telephones within the PBX may be connected to other sites in the same metropolitan area by a local Public Service Telephone Network (PSTN) carrier. The latter in turn may be interconnected to other metropolitan areas within a country by long distance or Wide Area telecommunications networks, which are in turn connected by communication channels operated by international carriers into a global telecommunication network.

Although the PSTN telecommunication network was originally designed to carry analog voice communications requiring only a bandwidth of about 4000 Hz for each conversation, telecommunication carriers learned early in the history of telephony that significant cost savings could be achieved by combining several telephone conversations and transmitting them over a single transmission channel, consisting of a single wire pair, for example. The process of combining multiple information signals such as those in multiple telephone conversations is referred to as multiplexing, while the process of recovering individual conversations from a common carrier signal and directing them to the proper destination telephone is called de-multiplexing.

While there are a variety of multiplexing and de-multiplexing techniques available, a method which is presently used most widely in the telecommunications industry is called Time Division Multiplexing (TDM). In TDM, analog information signals such as voice signals, are first digitized, i.e., converted into a stream of ones and zeros, or bits. The digital bits are then placed on a carrier signal such as an electrical current alternating at a frequency substantially greater than the maximum voice frequency which is to be transmitted, or on a laser beam, for example. This is done by modulating the carrier signal in unison with the sequential variations of ones and zeros in the information signal. Modulation consists of varying a characteristic such as the amplitude or phase of the carrier signal in

unison with the variation in ones and zeros of the information signal. In Time Division Multiplexing, the string of ones or zeros, called Packets, representing a particular telephone conversation, are interleaved, "or time sequenced," with packets of bits representing another telephone conversation, and transmitted on a common carrier signal. At the receiving end of the carrier signal, the packets of data representing the various conversations are split off from the other packets, converted into analog signals representing an original voice signal, and directed to the proper destination telephone.

Since PSTNs provide their typical subscribers with a telephone communication channel which has a bandwidth of 4 kHz, that channel may also be used to carry digital data signals, as long as the data bandwidth of the signals is within the allotted bandwidth. Thus, Modems (Modulators/Demodulators) are used to convert digital signals from telefacsimile machines and computers to packets of digital signals which may be transmitted over telephone lines. Accordingly, communications between individual computers and remote Internet sites are also routinely made over PSTN voice-quality lines. However, as can be readily understood, transmission of large amounts of data over reasonable time periods is frequently required by even modest sized enterprises. Therefore, telecommunications companies have made available wire or optical fiber communication lines which have a much greater bandwidth than ordinary voice grade telephone lines. For example, it is possible to rent T1 lines having a bandwidth of 1.544 Mbps (Megabits per second) in the United States, and E1 lines having a bandwidth of 2.048 Mbps in Europe. For enterprises requiring higher data transfer rates DSL (Digital Subscriber Lines) may be rented from the PSTNs, as can fiber optic lines having bandwidths ranging from several hundred Mbps, to several gigabits per second.

Not surprisingly, higher bandwidth communication lines are rented by the PSTNs at correspondingly higher prices. Moreover, as the following discussion will illustrate, the bandwidth requirements of even modest enterprise communications can be substantial. Thus, for example, a single voice grade digital telephone channel of the type connected to most residential telephones has a bandwidth of 64 Kbps (kilobits per record). This bandwidth requirement derives from the fact that ordinary voice communications, if they are to be transmitted with acceptable clarity and caller recognizability, must have, as stated earlier, a bandwidth of 4Khz, if transmitted as an analog signal. However, as is well known, the Nyquist sampling criterion requires that an analog signal must be

sampled at least twice the maximum frequency that is desired to reproduce. Accordingly, 4Khz voice signals must be sampled at 2X4Khz = 8Khz. Also, the dynamic range of voice signals required for acceptable communication has been determined to be about 256 to one, or 8 binary bits. Therefore, each digitized telephone connection channel must have a bandwidth of 64 Kbps. Thus, a T1 line, which at first glance would appear to have a substantially high bandwidth relative to that required for analog telephone conversations, can transmit only 24 digitized, TDM voice signals.

In addition to requiring substantial communication bandwidths for even modest numbers of telephone lines, most enterprises required substantially greater channel bandwidths for data interchange between enterprise sites and/or the Internet. Moreover, the increased use of video teleconferencing between various enterprise facilities requires even greater bandwidths. Thus, each time an additional group of telephones, new computer system, or video conferencing installation is added to an enterprise facility, it is generally required to procure additional communication lines from a PSTN service. This entails substantial capital investment and recurring costs, and the installation and connection of the new lines can disrupt enterprise operations.

In recognition of the problems resulting from increased communication channel bandwidths required by the burgeoning use of telephone, data, image and video transmissions by various enterprises, telecommunication experts have devised and implemented a mode of transmitting various signals of the foregoing type over a single communication channel. This technique is referred to as Asynchronous Transfer Mode.

To better understand ATM, and the novel advantages and benefits that the present invention contributes to ATM communications, it is perhaps useful to consider briefly data communication modes which preceded ATM. Thus, as described above, PSTN carriers transmit multiple voice signals over a single wire pair, optical fiber, satellite channel or the like, using time division multiplexing. In this communication mode, groups of individual bits, or packets, representing a single telephone conversation, for example, are interleaved in time with packets representing other conversations, into a single serial data stream. Typically, eight bits of information are grouped together in a serially arranged string to form an 8-bit Byte. Packets of bytes are then grouped together into a Frame, which adds a group of coding bytes called a header at the beginning of a data stream. Among

other things, the header identifies the source and destination addresses of information or PAYLOAD bytes which follow the header, i.e., arrive later. The length of a frame is not specified, but may be limited by a PSTN carrier to a maximum value, one thousand bytes, for example.

Since the length of a Frame is indeterminate in some instances, a trailer must be placed at the end of each Frame, indicating that the immediately preceding byte was the last byte in a payload, and indicating source and destination addresses of the next packet of bytes. This method of grouping bytes together and identifying source and destination addresses, as well as other parameters related to the intended disposition of a data stream, is referred to as FRAME RELAY and is widely and effectively used in the telecommunication industry.

By communicating information packets in Frame Relay Frames, computer files may be interleaved with telephone conversations and transmitted in the Frames. This interleaving may be optimized by utilizing Statistical Time Division Multiplexing (STDM). In STDM, pauses in certain communications which would normally be encoded into data packets that convey no information are replaced by data packets bearing useful information from another telephone conversation, computer data file or the like. The STDM technique works well enough with Frame Relay for interleaving certain types of data traffic, such as telephone conversations and computer data files, because the unpredictable interruption and resumption of computer data transfer is usually of no concern, as long as all of the data bits eventually arrive at their destination at an acceptable overall or average data rate. However, other types of data may not readily be interleaved in a Frame Relay Frame. For example, while an occasional interruption of data flow, or variable delays in the arrival of data at a destination generally are not problematic in the transfer of computer data, such interruptions or delays can cause video images to tear or otherwise degrade in an unacceptable fashion. Also, voice communications which are delayed more than about 100 mseconds can be a source of annoyance to persons engaged in a conversation, and CD quality, high fidelity sound is perceptibly degraded by delays or Latency Periods much greater than about 100 microseconds. Thus, the disparate bandwidths and delay requirements of voice, digital data, video, image, and music are relatively hard to reconcile using Frame Relay Multiplexing of such signals, and this difficulty motivated, at least in part, the creation of the Asynchronous Transfer Mode (ATM).

In ATM, each packet of bits representing information is defined as a CELL which has a length fixed at 53 eight-bit bytes, or octets. The first 5 bytes of each cell comprise a header which contains, among other things, information related to the source and destination of the 48-byte-payload which immediately follows the header. Since each cell is exactly 53 bytes long, it is generally not necessary to have a trailer indicating the end of a payload. Also, the header of each ATM cell contains information related to which Virtual Channel (VC) within a Virtual Path (VP) that the cell is to travel. Moreover, the Virtual Channel and Virtual Path taken by each cell is specified by Virtual Path Identifier and Virtual Channel Identifier bits, respectively, in the header, causing the cell to travel over a channel specified to afford a particular Quality of Service (QoS), which will now be explained.

There are presently five QoS categories in ATM, ranging from one accorded the highest network priority , for which a PSTN or other carrier generally charges the most, to the lowest network priority, which is generally the least costly. The highest QoS category is Constant Bit Rate (CBR), which is contracted for between a user and telecommunication carrier for sensitive applications requiring a constant throughput rate with minimal cell delays or loss. Applications requiring CBR include PCM (Pulse Code Modulated) data streams carrying real-time voice, video, and circuit emulation of private lines or other TDM circuits. The quality of service or QoS category having the second highest network priority is Variable Bit Rate-Real Time (VBR-RT) and is used for information which must be transmitted at a fairly predictable rate, and which is sensitive to delay and loss.

QoS service category 3 is called Variable Bit-Rate, Non-Real Time (VBR-NRT), and is used for information which is less sensitive to delays. QoS category 4 is called Unspecified Bit Rate (UBR), and is used for applications in which substantial delay times are tolerable. QoS category 5 is called Available Bit Rate (ABR) and is used for transmitted information that is less critical than UBR data.

ATM has proven to be a highly efficient data transmission protocol, and has therefore been adopted by PSTNs and other telecommunication carriers world-wide. These carriers have invested heavily in converting hardware and software systems which formerly could work only with the Frame Relay protocol, to systems in which ATM format signals can be Interworked, or transformed into Frame Relay signals, and *vice versa*. Computers used to direct ATM data streams to the proper destination

along wires, optical fibers or microwave carrier signals between ground stations or satellites are called Switches, and an ATM network whole is referred to as an ATM Backbone.

Devices which interconnect two or more networks are referred to as Bridges. Routers are devices which perform functions similar to those of Bridges, but function at a higher level. Thus, while a bridge knows the addresses of all the computers on each network joined together by the bridge, a Router also recognizes that other Bridges and Routers are on the network. Using that information, the Router is able to decide the most efficient path to send each message between a pair of end users. ATM networks may employ any of the devices described above.

A device of higher complexity than a Router exists, called a Gateway. The Gateway performs functions similar to that of a Router. However, in addition to routing functions, a Gateway is capable of translating or Interworking messages from one network format to the format of a different type of network. A Gateway can perform data format translations which enable data interchange between a LAN, such as an Ethernet LAN, and an ATM Backbone Network.

For an enterprise to fully exploit the advantages offered by ATM in achieving the goals of streamlining its communications while minimizing costs, it is usually necessary to have equipment on the enterprise site which enables the enterprise to connect its various systems to an ATM Backbone network. Such systems may include TDM voice signals from a PBX, video conferencing signals, Ethernet or other protocol LAN signals, among other types of data. ATM access equipment of this type are customarily referred to as Customer Premises Equipment (CPE), owing to the location of the equipment at an enterprise site. ATM CPEs provide a User to Network Interface (UNI), while interconnections between various nodes of an ATM network are called Netware Node Interfaces NNI).

There are presently available CPE devices which provide enterprises with access to an ATM Backbone network, thus allowing the enterprise to bundle its communications links, including voice, data, video and the like, onto a common communication channel. However, there are a number of problems with existing CPE devices affording ATM access. Such problems have limited the full utilization of the advantages offered by ATM.

Although problems associated with the enterprise utilization of ATM are diverse, a main source of problems is the inherent complexity involved in the segmentation of data cells received from a stream source, and the reassembly of cells from diverse downstream sources such as PBXs, LANs, video cameras and the like, into a single ATM cell stream. Thus, while the stripping of different serial data flows from incoming ATM cells into individual data flow queues, and the interleaving of various outgoing cell queues into a single ATM cell stream may seem to be a relatively straight forward task, it in fact requires substantially great real-time computing power. Of course, if one had a super computer available which is dedicated to the task of performing ATM access functions such as those of a Router or Gateway, the computational portions of these task functions may be readily performed. However, the various types of interfaces typically required of an ATM access device would still be problematic, even if the exorbitant cost of a super computer could be discounted.

Because of the inherent complexity involved in performing various functions required of ATM access devices, present devices fall into general categories: (1) Versatile and very expensive devices using raw, high speed computational power afforded by general purposes processors, and (2) Moderately priced devices having limited capabilities.

The present invention was conceived of to provide an Integrated Access Device for Asynchronous Transfer Mode (ATM) Communications, which provides a wide variety of CPE UNI functions with substantially greater proficiency than existing devices, and at a substantially lower cost. The foregoing advantages are achieved by the novel combination of a RISC (Reduced Instruction Set) processor with a custom PLA (Programmable Logic Array) or ASIC (Application Specific Integrated Circuit) having a variety of performance enhancing imbedded algorithms.

OBJECTS OF THE INVENTION

An object of the present invention is to provide an Integrated Access Device For Asynchronous Transfer Mode (ATM) Information communications, which provides bridging, routing, and interworking functions between ports selected from a group including ATM, Ethernet, Frame Relay, Voice, and Video signal technologies.

Another object of the invention is to provide an Integrated Access Device for ATM which converts incoming non-ATM signals to ATM signals, and imposes ATM QoS standards on the ATM

signals, thus allowing ATM QoS to be imposed on signals which may be inputted and outputted as non-ATM signals.

Another object of the invention is to provide an Integrated Access Device for ATM which provides ATM switching and scheduling utilizing a RISC microprocessor which is operably interconnected with a hardware programmed gate array so as to minimize computational and memory requirements of the microprocessor.

Another object of the invention is to provide an Integrated Access Device for ATM which utilizes a microprocessor operatively interconnected with a Programmed Gate Array via a local bus, and a plurality of expansion ports connected to the programmed gate array via an expansion port bus, whereby input/output modules of various types may be plugged into any of the expansion ports.

Another object of the invention is to provide an Integrated Access Device for ATM which utilizes a single functional block which serves as a scheduler to fully service multiple qualities of service (QoS).

Another object of the invention is to provide an Integrated Access Device for ATM which contains a functional block that assigns a scheduler resource to multiple ports in correct proportions, with fine granularity in representing relative rates and intervals.

Another object of the invention is to provide an Integrated Access Device for ATM which contains a functional block including a beaded buffer pointer chain with intermediate pointers, thereby enabling multiple processes queues to be combined into a single flow queue.

Another object of the invention is to provide an Integrated Access Device for ATM which contains a functional block that provides capabilities of cut-through routing of data streams through the device.

Another object of the invention is to provide an Integrated Access Device for ATM which contains a functional block that provides multiple preemptive CBRs for Precise Port Pacing Control.

Another object of the invention is to provide an Integrated Access Device for ATM that includes a functional block comprising a partitionable page shifter with self-timing XOR chain.

Another object of the invention is to provide an Integrated Access Device for ATM which includes a functional block that provides cell output flow rates having fractional interval times for fine granularity bandwidth allocation.

Various other objects and advantages of the present invention, and its most novel features, will become apparent to those skilled in the art by perusing the accompanying specification, drawings and claims.

It is to be understood that although the invention disclosed herein is fully capable of achieving the objects and providing the advantages described, the characteristics of the invention described herein are merely illustrative of the preferred embodiments. Accordingly, we do not intend that the scope of our exclusive rights and privileges in the invention be limited to details of the embodiments described. We do intend that equivalents, adaptations and modifications of the invention reasonably inferable from the description contained herein be included within the scope of the invention as defined by the appended claims.

SUMMARY OF THE INVENTION

SUMMARY OF THE INVENTION

The present invention is directed to an Integrated Access Device (IAD) supporting data and voice in the customer premise. The IAD is a 1U high chassis based product. A modular design will enable it to support several configurations.

The IAD main board contains all the circuitry and connectors for both the IAD application and the Fraim-IBM application and can be used in either product. The IAD is designed so that the form factor of the IAD main board is identical to the form factor of the Fraim-IBM CPU board.

The IAD is a functional bridge and IP router incorporating Ethernet, Frame Relay, ATM and voice technologies. With ATM switching and scheduling at its core, the IAD will fully support quality of service in ATM and be able to impose ATM QoS onto its non-ATM ports. It will support ATM PVC's and SVC's with UNI 3.0, 3.1 and 4.0 signaling. AAL-5 will be supported for data. The IAD will incorporate Frame Relay over ATM Interworking standards FRF.8 and FRF.5. AAL-1 and AAL-2 will be supported for voice. Both digital or analog voice will be supported.

The IAD can be modularized as shown in Figure 1. The Main Board performs the core ATM switching and scheduling functions and Frame Relay to ATM Interworking. The Voice Processor performs voice compression and conversion of TDM voice channels to AAL-1 or AAL-2. The other modules provide physical interfaces.

The Main Board has four expansion ports that connect to IAD input/output modules. Three of these apply to the IAD application However, it can be supplied with fewer or more IAD input/output modules.

The present Integrated Access Device (IAD) advances the state of the art with architecture that achieves unprecedented levels of performance and economy in the delivery of broadband services to branch and regional offices. Specifically, the IAD allows incumbent and competitive access providers to deliver REAL T1 multiservice access at a relatively low price.

The IAD defines a new class of access CPE, which delivers high-end performance at pricing that enables carriers to broadly offer integrated services to the branch office market segment. This is possible because of the IAD architecture which is a protocol interworking hardware accelerator that enables new levels of multiservice network processing capability in an economical, scaleable architecture.

The Challenge in Public and Private Networks

The task of bringing multiservice access to branch and regional offices presents unique challenges for equipment manufacturers:

1. *Cost of access bandwidth and equipment*. On a per-Mbps basis, low-speed access bandwidth is most expensive in the network because it has not benefited from technology investments like optical networking the WAN or gigabit Ethernet in the LAN.
2. *Limited bandwidth*. The vast majority of branch offices are still served by cooper. Although DSL technologies have made tremendous strides in increasing the usable loop bandwidth, it remains limited to a few Mbps or less.
3. *Price sensitivity*. Branch and regional office access is the most price sensitive networking segment. Even though these services are part of a large corporate IT budget, every dollar spent for branch access is multiplied by the number of branches in the corporate network, making price an important discriminator.
4. *Multiple communication protocols and traffic types*. Branch office IADs must be able to interwork between many communication protocols: Frame Relay, Ethernet, ATM, Internet Protocol (IP), digital time-division multiplex (TDM) voice, analog voice, T1/E1,

and xDSL. The complex translation process between these different protocols requires significant processor capabilities.

5. *Limited networking expertise at end-user*. The typical small, branch or regional office has little or no in-house networking expertise.

When the technical requirements placed on IADs -- easily managed platform with support for multiple protocols, bandwidth maximizing capabilities and robust traffic management -- are compared to the cost sensitivity of the branch office access market segment, it quickly becomes apparent that IADs present one of the most challenging design problems in networking.

In the past, access service providers could choose between two types of IADs for service deployment: high-end, high-performance equipment designed to scale to OC12 speeds, but not cost effective at T1 or multi-T1 speeds; or low-end, low-cost microprocessor-based equipment that have difficulty operating at wire speed when faced with a random mix of protocols.

A Better Solution

The IAD hardware-based networking processing accelerator specifically addresses the needs of the branch office access marketplace. The IAD hardware-based network processing accelerator operates in conjunction with a cost-effective RISC processor. Microprocessors are very effective in performing configuration and management functions but not efficient with highly repetitive data forwarding functions. The IAD hardware-based accelerator serves as the data forwarding engine, resulting in a high performance partnership. However, because the hardware-based accelerator is optimized for the branch office access challenge, it remains a very cost-effective solution.

At the core of the IAD hardware-based accelerator is an ATM switch and traffic shaper. This is surrounded by a protocol-interworking machine, allowing the hardware-based

accelerator to adapt any type of traffic (TDM, IP, or Frame Relay, for instance) to ATM, apply ATM quality of service (QoS) to the traffic, then adapt it back into any other protocol. In this way, the hardware-based accelerator can provide any data flow with robust ATM QoS, even if the flow enters and exits the IAD in some other protocol.

The IAD performs various protocol tasks, like Ethernet bridging, IP routing, and Frame Relay-to-ATM interworking, while optimizing the traffic characteristics of the data flows. The tight coupling between the IAD hardware-based accelerator and the RISC processor also enables the IAD's performance to scale to meet the future needs of the branch office. The IAD applies ATM inverse multiplexing to aggregate several wideband links into a single braodband connection, allowing carriers to deliver more services over existing copper plant rather than waiting for a slow fiber build-out program. Alternative access providers (wireless local loop, point-to-point radio, digital cellular radio, etc.) can also take advantage of the IAD's IMA technology since it is transparent to the physical layer employed.

To take advantage of this protocol flexibility, the IAD can be built as a modular chassis, allowing carriers to customize the platform for their particular networks' and markets' needs, such as the following interfaces: Ethernet, synchronous serial, quad T1 ATM IMA, and digital T1 PBX interfaces.

In the modern networked enterprise, information technology must reach the most remote corner of the enterprise -- however, this must be achieved without a similar deployment of network support personnel. The AID has been designed to meet these goals, including comprehensive remote management that allows configuration, monitoring and control without a truck-roll or site visit.

The IAD represents a major step forward in the provisioning of true broadband services to small, remote branch office locations.

For the customer, the IAD enables the realization of the true connected enterprise where discrimination based on location can become a thing of the past.

For the service provider, it allows them to capture the super-valuable business broadband service mark opportunity, without having to wait for fiber deployment.

For the alternative access provider, the IAD IMA technology, in combination with rapid-deployment wireless technologies, unlocks new opportunities. The IAD allows rapid capture of high-value business markets without the need for capital investment in fixed local loop transmission technologies.

The IAD represents a step change in opportunity. It opens new horizons in broadband deployment to the very edge of the enterprise or network by using the infrastructure which is already sitting in the ground -- across the nation and the world.

<u>Overview</u>

The IAD of the present invention is an Integrated Access Device (IAD).

The IAD is a Customer Premises Equipment (CPE) solution that enables organizations to connect multiple branch offices economically to a multiservice ATM or Frame Relay Wide Area Network (WAN). It provides the means for branch end-users to combine their voice and data network connections on to a single low-speed network path, which can be more easily managed from the central headquarters.

The IAD connects to the customer's existing data, voice, and video equipment and resides in the end-user's communications room or closet. It is a sophisticated, branch-office, multiservice platform that provides many additional key functions and benefits over other CPE devices such as Frame Relay Access Devices (FRADs) or Time Division Multiplexers (TDMs).

The IAD can be configured as a host or CPE access device to provide:

1. Frame Relay to ATM interworking;

2. Inverse Multiplexing over ATM (IMA) for up to 4 x E1/T1 lines;

3. Variable Bit Rate voice adaptation using AAL2 protocols;

4. Circuit Emulation Services using AAL1 protocols;

5. Comprehensive support for voice compression modulations;

6. Echo cancellation and silence suppression for AAL2 protocols;

7. Attachment to digital PBX using E1/T1 interfaces;

8. Analogue FXO (Foreign Exchange Office) and FXS (Foreign Exchange System) operation with Ground or Loop Start;

9. E&M (Electrical and Mechanical tie line) support for Types 1, 2, and 5 (Immediate, Delay, and Wink);

10. Support for voice, video, or data over single or multiple ISDN-BRI;

11. IP routing and bridging over ATM;

12. DHCP (Domain Host Configuration Protocol) and NAT (Network Address Translation) support;

13. Comprehensive support for SNMP network management;

14. Maximum of 4096 connections (FR DLCIs (Frame Relay Address), ATM VCs, etc.);

15. ATM PCR (Peak Cell Rate), SCR (Sustained Cell Rate), and MBS (Maximum Burst Size) traffic shaping;

16. ATM classes: CBR, VBR-rt, VBR-nrt, UBR and UBR+;

17. ATM PVCs and SVCs;

18. Per port pacing;

19. Frame Relay QoS via DLCI : CIR; and

20. Conformance to ATM and Frame Relay forum standards.

Interworking Technology

The IAD interworking solutions provide peer-to-peer connectivity between the IAD located in the branch offices and IADs located in the central or regional office locations. ATM or Frame Relay PVCs or are mapped according to networking requirements, which provide for a fully meshed configuration to exist between all IADs within a given Multiservice WAN.

Inverse Multiplexing over ATM

The IAD offers the capability of connecting up to 4 x 2Mbps circuits into a logical IMA group, thus allowing ATM PVCs or SVCs to utilize available bandwidth fully. In this mode, the IAD connects to the ATM WAN switch via multiple 1.5Mbps (T1) or 2Mbps (E1) leased lines. The adjacent ATM switch must be configured with an equal IMA facility to terminate the logical group prior to core network switching of cell traffic, or the IMA group can be carried intact across the WAN to another IAD for termination.

Enhanced Voice Convergence

The IAD supports the multiplexing of compressed voice channels via ATM Adaptation Layer 2 (AAL2) protocols into a single ATM PVC or SVC, thus maximizing ATM bandwidth optimization. Further bandwidth efficiencies are obtained through utilizing silence suppression algorithms and local comfort noise generation to eliminate unnecessary cell transmissions. Additionally, the IAD supports uncompressed voice channel transmission via AAL1 structured Circuit Emulation Services (CES) to an adjacent IAD or other vendor equipment.

IP Routing and Bridging

The IAD offers unparalleled performance versus cost using its proprietary technology to perform frame to cell conversion and data forwarding in hardware. The IAD performs both local IP routing (RIPv1 & v2) and switching as well as ATM bridging using multi-protocol encapsulation techniques over AAL5 (RFC 1483 and RFC 1577 for Classical IP). The bridging function also supports the Spanning Tree protocol.

Frame Relay to ATM Interworking

Local data connections are managed via the IAD's Frame Relay to ATM Interworking function. This facility enables customers to retain their existing router hardware and software configurations to preserve access to legacy applications. The data connection operates up to 2Mbps via a DB25 V.35X.21RS-530, or RS-449 interface. The interworking function supports either Network (FRF.5) or Service (FRF.8) Interworking in accordance with the Frame Relay Forum multi-protocol implementation agreements (RFC 1490

ATM Classes of Service

ATM PVCs and SVCsare fully supported to ATM UNI 3.0, 3.1, and 4.0 signaling. Quality of Service and traffic shaping per port is provided via VCC PCR, SCR, and MBS parameters. Service classes are supported via Adaptation Layers 1, 2, and 5 utilizing classes CBR, VBR-rt, VBR-nrt, UBR, and UBR+.

Advanced Network Management

The IAD provides extensive network management facilities via its internal SNMP agent and a supporting SNMP Network Management Application. A full range of functions is available to configure, monitor, and report upon network performance, configuration parameters, call management, fault management, and IP/Frame Relay network protocol statistics.

The IAD Management

Management of IAD is available through local and remote access to one or more IAD's via SNMP. The application is designed to provide the network management capabilities expected from enterprise or carrier-class customers. Network management is generally defined to encompass two main areas, namely Monitoring and Control. Preferably, the IAD management can be through Mariner Networks, Inc., Anaheim, California, Messenger™, SNMP Network Management Application which can provide local and remote access to one or more of the IAD's.

Network Monitoring is concerned with observing and analyzing the status and behavior of its network domain configuration and its devices.

Network Control is concerned with the altering of parameters of various configurations of the network devices and causing those components to perform predefined actions.

In line with this concept, IAD is a fully managed ATM IAD, which supports the following key disciplines:

1. Network Management,
2. Traffic Management,
3. Code Management,
4. Security Management.

Network Management

The IAD's subsystems can be managed in any of the following ways:

From an ASCII terminal with a character-based command line interface that is directly connected to the console monitor port.

By remotely logging into a command line interface via a Telnet session. This session may be via the local Ethernet port, Frame Relay port, or in-band across the ATM WAN.

By accessing the IAD's SNMP Agent via an authorized network management station, such as a station running Mariner Networks' SNMP Management Application "Messenger". The network management station may reside anywhere in the network.

The IAD's Messenger™ application can be run on any type of network management workstation irrespective of operating system or machine type. It can be run under HP OpenView™ or independently, offering a complete network management environment for the enterprise or carrier class user. The graphical user interface (GUI) enables the operator to configure the IAD elements quickly and easily and to interrogate performance data and traffic

profiles in a variety of tables and charts. Multiple IAD configurations and maps may be viewed simultaneously.

The IAD can support simultaneous access by multiple network management stations to facilitate redundancy and continuous network operational requirements. The SNMP agent can comprise Mariner Networks' Enterprise MIB and a number of industry compliant networking MIBs (ATM, FR, and MIB-II).

Traffic Management

The IAD's advanced traffic management functions include:

1. Priority queues per ATM Quality of Service (QoS),
2. Constant Bit Rate (CBR),
3. Real time Variable Bit Rate (VBR-rt),
4. Non-real time Variable Bit Rate (VBR-nrt),
5. Unspecified Bit Rate (UBR)
6. Unspecified Bit Rate Plus (UBR+), and
7. Traffic shaping per port and per Virtual Circuit (TM 4.0).

The IAD ensures that the Virtual Channel Connection (VCC) contract is respected at the Virtual Channel (VC) level. To reduce irregular bursts of traffic, a reshaping function is provided.

Code Management

Code management allows the network administrator or network operator to manage the application and user configuration modules contained within the IAD. The application module contains the program logic necessary for the IAD to function. User configuration modules consist of parameters and network definitions that describe the network, voice characteristics, profiles, and packet/cell routing information.

The IAD's flash memory can hold multiple copies of application modules as well as multiple copies of user configurations, and allows an operator to switch between them. In this

way, the IAD can be reloaded or re-configured to perform differently while still retaining the ability to recover from updates that fail to function as required.

IAD's code management can be accessed in any of the following ways:

1. Application and user configuration module data can be uploaded or downloaded using TFTP (Trival File Transfer Protocol). The IAD contains a TFTP server that enables bi-directional processes.
2. Switching between application or user configuration data can be performed using either the console port via the command line interface (CLI), via a Telnet session, or remotely via the Management application.
3. Using the console monitor port, uploading and downloading of application or user configuration data can be performed.

Providing multiple copies of application and user configuration data in flash memory enhances the IAD's network manageability in a customer premises environment. The IAD's advanced network management capabilities enable network control and monitoring to be performed quickly and simply with the minimum of end-user involvement.

Security Management

The IAD can be configured with the following security features:

Configuration Protection

Access to the IAD via the console monitor port can be password protected to protect the IAD's configuration. This password can be changed at the customer's/end-user's discretion. A hardware-based reset feature can be incorporated to enable recovery to a default password in the event of password loss.

Network Access Protection

Telnet access to the IAD's Command Line Interface (CLI) via the ATM, local Ethernet or Frame Relay network is provided and access is controlled via a password.

Access to the IAD SNMP agent is controlled via a domain name to prevent and limit unauthorized use.

Typical Implementations

The IAD simplifies ATM access at the customer premises. This is achieved through implementing the IAD as an ATM Interworking Network Terminating Unit (NTU) that clearly defines the boundary of the ATM network from the customer's local network communications equipment. Through its ATM interworking capabilities, the IAD converges multiple services (voice, data, and video) over single or multiple upstream ATM links. Figure 2 illustrates a typical configuration.

Figure 11 illustrates a simple "mesh system" implemented between several office locations. All IADs are configured to establish PVCs (Permanent Virtual Circuits) between remote locations and to the central location housing the host system and application servers. Multiple IADs may be installed at the central location to provide sufficient voice channel capacity for head office personnel.

The IAD product can consist of a multi-slot, such as a 3-slot, chassis enclosure with the following components:

1. Main processor board with application software loaded,
2. Power supply assembly,
3. 1 x RJ45 Ethernet port,
4. 1 x DB9 RS-232 console monitor port, and
5. Three or more blank single-slot filler plates.

The following components can be furnished with the IAD to facilitate power up and initial configuration:

1. Power supply cord,

2. RS232 modem cable, and

3. Documentation CD-ROM package.

<u>**System Component**</u>

All IAD units are based upon a main processor board design and chassis enclosure that facilitates the insertion of one or more Network or User Interface Modules depending upon the number of available slots. The modules are described below.

The main processor board contains the CPU, various memory modules, operating system, and application code. Additionally, this board holds a switch processor, either a programmable logic device or an ASIC that performs frame to cell conversion and data forwarding in hardware.

The IAD can be equipped with a single RJ45 socket on the front panel system unit to facilitate either 10BaseT Ethernet or Telnet management access. In this way, the IAD can be configured without the need for any modules to be inserted prior to use. Initial configuration of IP (Internet Protocol) addressing would need to be achieved via the console monitor port.

The IAD is preferably equipped with the DB9 RS-232 female DCE connector unit to facilitate initial configuration of the IAD unit.

Main memory is provided in all IAD configurations. In addition to this memory offering, IAD is configured with flash memory to hold multiple application and user configuration data, and boot PROM to support initial power-on and program load functions. Sixteen (16) MB of DRAM memory can be used; more or less memory can be used.

Each unit is preferably configured with an internal auto-detecting VAC power supply with a fused power switch and a power cord.

A printed Quick Start Installation Guide is preferably provided with all IAD units. All other documentation relating to IAD is available on an accompanying CD-ROM or other memory

device or on a website. Additionally, all user-related documentation is available by downloading from the Mariner Networks website.

Each IAD is fitted with a modem cable, such as an RS-232 DB9 DCE/DTE modem cable. Access to the console monitoring port is through a terminal device, such as a VT100 terminal device.

In addition to the base components supplied with the chassis, the IAD will need to be populated with one or more Network or User Interface Modules that connect the ATM WAN or the existing customer communications equipment.

The IAD is preferably designed to be either a standalone, wall-mounted, or rack-mounted unit. Mounting kits can be made available to facilitate the installation of the IAD into a 19 inch communications rack or onto a wall.

A number of cabling options are preferably supported to accommodate connection of the ATM interface, and Frame Relay V.35/X.21 attached router to the IAD.

The IAD can be supported by many types of network modules and user modules (Interfaces) which can be adapted to be received in universal slots, i.e. slots that will accept modules of any type of interface protocol, or dedicated slots that will receive a more limited number of modules of specific types of interfaces protocols. Universal slots are preferred. A limited number of network and user modules are identified in Table 1.

Network Module

A 1 x port T1/E1, or 4 x port T1/E1 module can be provided.

Each module may be configured to operate in ATM cell delineation or Frame Relay HDLC delineation mode. Each interface can be presented as an RJ48C female socket that can accept either a T1 (1.5Mbps) or an E1 (2Mbps) facility interface.

Each module can have the following characteristics:

1. 1 or 4 ports each operating at either 1.544Mbps or 2.048Mbps line rate.
2. Each port may connect to an ATM switch via UNI (3.0, 3.1, or 4.0), or a Frame Relay DLCI compliant device.
3. Integrated CSU/DSU functionality.
4. Physical interface is electrical with impedance of 100/120 Ohms.
5. One or more modules may be inserted into the IAD depending upon the available slots.
6. Both modules are preferably easily swappable without the need for specialist knowledge or equipment. The IAD will probably require rebooting and reconfiguring upon change of module type.

Figure 12 shows a 1 x port T1/E1 and 4 x port T1/E1 module face plates.

<u>**Network Module**</u>

A 4 x port E1 or T1 module for ATM Inverse Multiplexing over ATM (IMA) network can be provided.

This module may be configured to operate in a variety of logical IMA line groups. Each interface can be presented as an RJ48C female socket that can accept either a T1 (1.5Mbps) or E1 (2Mbps) facility interface.

The module has the following characteristics:

1. 4 ports, each operating at either 1.544Mbps or 2.048Mbps line rate.
2. Each port may connect to an ATM switch via UNI (User Network Interface) using a supported interface.
3. T1 option has an integrated CSU/DSU (Channel Service Unit/Data Service Unit) functionality.
4. Physical interface is electrical with impedance of 100/120 Ohms.
5. One or more modules may be inserted into any of IAD's slots.

6. This module is preferably easily swappable without the need for specialist knowledge or equipment. IAD will probably require rebooting and reconfiguring upon change of module type.

Figure 13 shows the faceplate of the module.

Network Module

A 1 x port DS-3 or 1 x port E3 network module for ATM DS-3/E3 network can be provided.

Each module can be configured to operate in ATM cell delineation mode. Each interface is preferably presented as a BNC 75 Ohm female connector that can accept either a DS-3 (45Mbps) or an E3 (34Mbps) facility interface.

Each module preferably has the following characteristics:

1. 1 port operating at 34Mbps or 45Mbps line rate.
2. Each port may connect to an ATM switch via UNI using a supported interface.
3. Physical interface is electrical with an impedance of 75 Ohms.
4. One or more modules may be inserted into any of the IAD's slots depending upon availability.
5. Both modules are preferably easily swappable without the need for specialist knowledge or equipment. IAD will probably require rebooting and reconfiguring upon change of module type.

Figure 14 shows the faceplate of the module.

Network Module

A 1 x port OC-3 or 1 x port STM-1 for ATM OC-3/STM-1 network can be provided.

Each module is configured to operate in ATM cell delineation. The interface is presented as an optical fiber ST female connector that can accept either an OC-3 (155Mbps) or STM-1 (155Mbps) facility interface.

The module has the following characteristics:

1. 1 port operating at 155Mbps line rate software configurable between either the OC-3 or STM-1 format.
2. Each port may connect to an ATM switch via UNI using a supported interface.
3. Physical interface is single or multimode optical fiber.
4. One or more modules may be inserted into any of IAD's slots depending upon availability.
5. This module is easily swappable without the need for specialist knowledge or equipment. The IAD will probably require rebooting and reconfiguring upon change of module type.

Figure 15 shows the faceplate of the module.

Network Module

A 2 x port SDSL network module for the SDSL network can be provided.

The module may be configured to operate in ATM cell delineation or Frame Relay delineation mode. The module may be configured to communicate with another IAD, DSLAM or other Central Office (CO) equipment. The module can be configured as either a CO or CPE (Customer Premises Equipment) device.

The module has the following characteristics:

1. 2 ports operating in variable rate SDSL (symmetric Digital Subscriber Line) using Globspan s"!2B1Q X DSL chip set. SDSL data rates of 144kb/s, 272kb/s, 400kb/s, 528kb/s, 784kb/s, 1040kb/s, 1168kb/x, 1552kb/s, 2064kb/s, and 2320kb/s are supported using 2B1Q line encoding data rates.
2. Each port may connect to an ATM switch via UNI, or a Frame Relay compliant device.
3. Physical interface is electrical with impedance of 50/75 Ohms. The connectors are RJ11 terminating voice grade telephone wire local loops.

4. One or more modules may be inserted into any of IAD's slots depending upon availability.

5. This module is easily swappable without the need for specialist knowledge or equipment. IAD will probably require rebooting and reconfiguring upon change of module type.

Figure 16 shows the faceplate of the module.

Network Module

A 1 x port ATM/FR for HDSL2 network can be provided.

The module may be configured to operate in ATM cell delineation or Frame Relay delineation mode. The module may be configured to communicate with another IAD, DSLAM (Digital Subscriber Line Access Multiplexer), or other Central Office (CO) equipment. The module can be configured as either a CO or CPE device.

The module has the following characteristics:

1. 1 port operating up to 1.5Mbps using 2B1Q line encoding data rates.
2. The port may connect to an ATM switch via UNI, or a Frame Relay compliant device.
3. Physical interface is electrical with impedance of 50/75 Ohms. The connector is RJ11 terminating voice grade telephone wire local loops.
4. One or more modules may be inserted into any of IAD's slots depending upon availability.
5. This module is easily swappable without the need for specialist knowledge or equipment. IAD will probably require rebooting and reconfiguring upon change of module type.

Figure 17 shows the faceplate of the module.

Port Module

A 1 x port user or network module for synchronous serial lines can be made available.

The module is configured to operate in Frame Relay mode, clear channel or channelized mode, or ATM mode via clear channel. The module can attach to an existing Frame Relay router or other Frame Relay compliant device. The interface can be configured for either V.35 or X.21 via an adapter cable..

The module has the following characteristics:

1. 1x DB25 female DCE/DTE synchronous port supporting, RS-530, or RS-449. Data rate can be set from 64K to 8.192Mbps, full duplex operation.
2. One or more modules may be inserted into any of IAD's slots depending upon availability.
3. The module is easily swappable without the need for specialist knowledge or equipment. The IAD will probably require rebooting and reconfiguring upon change of module type. Figure 18 shows the faceplate of the product guide.

User Module

A 4 x port 10/100BaseT user module can be made available.

The module is configured to attach to an existing Ethernet LAN via a hub or switch. Each RJ45 port is rate auto-sensing and provides either switching of Ethernet packets between IAD's LAN interfaces or routing/bridging via AAL5 encapsulation over the ATM WAN.

The module has the following characteristics:

1. 4 ports of 10/100BaseT for local Ethernet or Telnet management access.
2. Spanning Tree protocol is supported.
3. Each port is on its own segment.
4. One or more modules may be inserted into any of IAD's slots depending upon availability.

5. The module is easily swappable without the need for specialist knowledge or equipment. IAD will probably require rebooting and reconfiguring upon change of module type. Figure 19 shows the faceplate of the module.

User Module

A 1 x port T1/E1 user module for voice T1/E1/PRI can be made available.

The module may be configured to operate in either T1 or E1 mode and connects to the customer's local PBX system. The module provides a T1/E1 trunk type interface that can support either 24 (T1) or 30 (E1) channels of voice throughput. PBX supported interface signaling includes either Robbed Bit (T1), CAS (E1), or ISDN PRI using Common Channel Signaling (CCS) to provide 23 (T1) and 30 (E1) bearer channels respectively for voice trunking. The module also contains the necessary Digital Signal Processors (DSPs) and logic to provide voice compression, silence suppression, echo cancellation, AAL1AAL2 processing, and packet to cell conversions.

The module has the following characteristics:

1. 1 port operating at either 1.544Mbps (T1) or 2.048Mbps (E1). The module can be ordered with support for 8, 16, 24, or 32 voice channels. These channels may be assigned to any time slot in the T1 or E1.
2. Signaling supported includes RBS, CAS (E1) and ISDN PRI (CCS).
3. Supported CCS signaling for ISDN PRI includes PRI Net5 User, PRI Net5 Network, and PRI QSIG.
4. AAL1 voice processing in accordance with af-vtoa-0078.000.
5. AAL2 voice processing in accordance with ITU-T I.363.2.
6. Voice processing includes G.711 (64K PCM), G.726 ADPCM, G.727 EADPCM, G.729 CS-ACELP, G.729AB CS-ACELP, and G.723.1A.

7. Support for Fax Relay and voice-band signaling.
8. Physical interface is an RJ45 electrical with impedance of 100/120 Ohms.
9. One or more modules may be inserted into any of IAD's slots depending upon availability.
10. The module is easily swappable without the need for specialist knowledge or equipment. The IAD will probably require rebooting and reconfiguring upon change of module type. Figure 20 shows the faceplate of the module.

User Module

A 1 x port T1/E1 + 1 x port ISDN BRI user module for voice can be made available.

The PBX T1/E1 facility interface operates identically as outlined for the previous user module. Additionally, this module incorporates an ISDN BRIport that provides for attachment to a videoconferencing codec (although it may be used with any ISDN BRI compliant device).

The module has the following characteristics:

1. 1 port operating at either 1.544Mbps (T1) or 2.048Mbps (E1). The module can be ordered with support for 8, 16, 24, or 32 voice channels.
2. Identical characteristics to that of the PBX E1/T1 module.
3. 1 ISDN BRI port providing 2 x 64K bearer channels and 1 x 16K D channel. Both S/T and U interfaces are supported.
4. One or more modules may be inserted into any of IAD's slots depending upon availability.
5. The module is easily swappable without the need for specialist knowledge or equipment. The IAD will probably require rebooting and reconfiguring upon change of module type. Figure 21 shows the faceplate of the module.

User Module

A 2 x port ISDN BRI or 3 x port ISDN BRI user module for integrated service digital network can be made available.

This module is equipped with either a dual port or triple port ISDN BRI facility that supports S/T and U interfaces. Each port can be configured to support voice, fax, or voice-band data signals. Full voice processing is supported for compressed or uncompressed transmission across the ATM WAN.

Each version of the module has the following characteristics:

1. 2 or 3 ports providing ISDN BRI service. Each port supports 2 x 64K bearer channels and 1 x 16K D channel. Both S/T and U interfaces are supported.
2. One or more modules may be inserted into any of IAD's slots depending upon availability.
3. Both modules are easily swappable without the need for specialist knowledge or equipment. The IAD will require rebooting and reconfiguring upon change of module type.

Figure 22 shows the faceplate of the module.

In one embodiment, the IAD comprises a main processing board that contains core memory, application code, and optional interface modules. A key element of this design is the ATM switch processor.

The ATM switch processor consists of a cell switching fabric with segmentation and re-assembly processes and a cell forwarding architecture that includes a cell scheduler function. It contains the necessary logic and dynamic tables to translate between ATM VCs and Frame Relay DLCIs. Additionally, through its powerful scheduling ability, it supports current ATM and Frame Relay Quality of Service (QoS) attributes. The processor uses an on-board CPU to build and maintain its tables and routing information.

The ATM switch processor's unique benefit is that once its tables have been defined, it converts, routes, and switches frames and cells effortlessly, in hardware, and releases the main CPU to perform other processor intensive tasks such as voice processing. Unlike other comparable CPE devices, this blend of technology enables the IAD to deliver the processing power and switching performance that would normally be found in larger and more expensive access units.

The IAD's other key components are the following subsystems:

1. ATM Processing,
2. Voice Processing,
3. Network Management.

The ATM Processing subsystem provides the broadband services to IAD's applications.

Overview

ATM processing, frame to cell conversion and transmission of cells to the ATM network modules is performed by the ATM switch processor.

The following ATM Adaptation Layers (AAL) and associated service classes are supported:

Layer	Service Class	Mnemonic
AAL1	Constant Bit Rate	CBR
AAL2	Variable Bit Rate	VBR-rt VBR-nt
AAL5	Unspecified Bit Rate	UBR UBR+

Table 2.
Supported AAL Protocols

AAL1 Operation. This layer is used to support all switched or permanent uncompressed voice calls. Uncompressed voice traffic is either carried as a structured or basic Nx64K CES cell stream as defined in the af-vtoa-0078.000 interoperability specification, Circuit Emulation Services (v2).

AAL2 Operation. This layer is used to support all switched compressed voice calls over the ATM network. All AAL2 voice traffic between a pair of IADs is multiplexed across a single ATM VC.

AAL5 Operation. This layer is used to support all Frame Relay data frames and Internet data packets over the ATM network.

Quality of Service. The IAD performs traffic shaping of its outgoing ATM cell flow in accordance with the relevant standard for Connection Traffic Descriptor that was negotiated with the ATM network. The relevant parameters used to specify unambiguously the conforming cells of the ATM connection are Peak Cell Rate (PCR), Sustainable Cell Rate (SCR), and Maximum Burst Size (MBS). IAD contains two leaky buckets to support its QoS scheduling.

Inverse Multiplexing over ATM Interface. The IAD can be configured to accept 2Mbps circuits via a 4-port E1/T1 IMA interface, which can be configured into two IMA logical

groups. Typically, ATM PVCs would utilize all available circuits in the IMA group to provide greater throughput. An outline flow of ATM cells through an IMA configuration is illustrated in Figure 23. Here, an ATM data stream is split across three individual physical links on a cell-by-cell basis in a "round-robin" effect.

<u>Frame Relay to ATM Operation</u>. The IAD supports both Frame Relay to ATM "Network" and "Service" interworking as defined by the Frame Relay Forum's Frame Relay/ATM Network and Service Interworking Implementation Agreements (FRF.5 and FRF.8 respectively).

<u>Network Interworking</u>. This function is responsible for forwarding frames between the Frame Relay interface and the ATM Data Subsystem. The IAD processes frames received from the Frame Relay interface as follows:

1. De-multiplexed according to their DLCI.
2. Stripped of their HDLC encapsulation headers.
3. BECN (Backward Explicit Congestion Notification), FECN (Forward Explicit Congestion Notification), and DE (Disregard Eligibility) congestion and flow control indicators are mapped according to ATM EFCI (Explicit Forward Congestion) and CLP (Cell Loss Priority) settings.
4. Re-encapsulated in ATM AAL5 CPCS PDUs.
5. Segmented and multiplexed over the UTOPIA (Universal Test and Operations Interface for ATM) cell interface according to the ATM VCC (Virtual Channel Connection).

In the reverse direction, the ATM cell traffic is processed as follows:

1. ATM AAL5 CPCS PDUs (Protocol Data Unit) reassembled from the UTOPIA cell interface.
2. De-multiplexed according to the ATM VCC.

3. Stripped of their AAL5 encapsulation overhead bytes.

4. ATM EFCI, DE congestion, and flow control indicators are mapped according to FR BECN, FECN, and DE settings.

5. Multiplexed over the appropriate Frame Relay interface according to DLCI.

Figure 24 illustrates Network interworking mapping performed between frames and cells.

<u>**The Service Interworking (FRF.8)**</u>. This function is essentially the same as the previous network function, except that protocol conversion algorithms are applied to convert Frame Relay bridged or routed PDU to ATM bridged or routed PDUs. Frames received from the Frame Relay interface are processed as follows:

1. De-multiplexed according to their DLCI.

2. Stripped of their HDLC encapsulation headers.

3. Network protocol encapsulation headers mapped from those specified in RFC 1490 (for Frame Relay) to those specified in RFC 1483 (for ATM).

4. Re-encapsulated in ATM AAL5 CPCS PDUs.

5. Segmented and multiplexed over the UTOPIA cell interface according to the ATM VCC.

In the reverse direction, the IAD processes the ATM cell traffic as follows:

1. ATM AAL5 CPCS PDUs reassembled from the UTOPIA cell interface.

2. De-multiplexed according to the ATM VCC.

3. Stripped of their AAL5 encapsulation overhead bytes.

4. Network protocol encapsulation headers mapped from those specified in RFC 1483 (for ATM) to those specified in RFC 1490 (for Frame Relay).

5. Multiplexed over the appropriate Frame Relay interface according to DLCI.

Figure 25 illustrates Service Interworking mapping performed between frames and cells.

Ethernet Operation

The IAD is assigned an IP address and subnet mask to each network port (including ATM WAN ports). Services such as Domain Host Control Protocol (DHCP) and Network Address Translation (NAT) are supported.

The IAD performs both local IP routing (RIPv1 & v2) and switching between its local and network ports. Bridging between a pair of IADs is achieved by using ATM bridging multi-protocol encapsulation techniques over AAL5 (RFC 1483) and Classical IP encapsulation (RFC1577).

Other protocols built into the IAD IP stack include the following protocols: UDP, TCP, TFTP, SNMP, ARP, and ICMP. Telnet packets received from the local ports or via the network ports are converted to command strings and passed to the IAD's command line interface (CLI) for parsing.

Domain Host Configuration Protocol

The Dynamic Host Configuration Protocol's (DHCP) purpose is to enable individual computers on an IP network to extract their configurations from a server (the 'DHCP server') or servers, and in particular, servers that have no exact information about the individual computers until they request the information. The overall purpose of this is to reduce the work necessary to administer a large IP network. IAD contains a DHCP server function

Network Address Translation (NAT) is used to translate one IP address to another. NAT can be used to allow multiple PCs to share a single Internet connection. It can also be used as a security tool by shielding the IP addresses of devices within the attached intranet. NAT can also be used for general IP address management by protecting the attached intranet from excessive address changes due to other network addressing constraints.

Voice Processing. This subsystem provides the voice and video-oriented narrowband services to the IAD's applications.

This section describes the functional aspects of IAD's voice processing capabilities. the IAD's voice traffic across the ATM WAN is managed using a mixture of both AAL1 CBR connections and AAL2 VBR-rt connections.

AAL1 is used to carry uncompressed voice channels and associated Robbed Bit or CAS signaling transparently, end-to-end. AAL2 is used in conjunction with a signaling and compression engine such as Mariner Networks' proprietary signaling and compression engine, to switch and carry packetized, compressed voice traffic end-to-end. The AAL type is software configurable on a trunk channel basis, and compression algorithm/ratio basis.

The IAD utilizes structured Circuit Emulation Services (CES), nailed up circuits supporting Nx64K (uncompressed) between IADs, or between the IAD and other vendors' equipment supporting standards-based CES. While uncompressed CES-based connections are less efficient than compressed, AAL2 based connections, they offer the greatest benefit in terms of end-to-end voice quality and interoperability.

Figure 26 illustrates some of the network interconnection scenarios that can be implemented using structured circuit emulation with a IAD network.

In Figure 26, each of the ATM PVCs shown (A, B, C) carries a fixed, constant bit rate stream of ATM cells. The cell payloads, formatted according to the rules specified in af-vtoa-0078.000, contain voice samples and robbed bit signaling information for the trunk channels that the associated PVCs are configured to transport between the attached voice interfaces and the ATM network.

A CES connection provides a "nailed-up" transport for TDM voice data and voice signaling, allowing geographically dispersed telephony endpoints to communicate transparently over the ATM network.

Circuits can be configured for either "Basic Mode", meaning that trunk channels are transported without associated signaling, or CAS mode, meaning that CAS/robbed bit signaling

information is included in the cell payloads. The latter is useful for connecting non-PBX type equipment (e.g., analog handsets) at one end to PBX/trunk terminating equipment at the other end (loop extension).

Compressed Voice Services

By using AAL2 VBR-rt ATM circuits in conjunction with IAD's compression and signaling software, IAD can more efficiently transport voice and fax traffic across the ATM WAN.

AAL2 provides for the bandwidth-efficient transmission of low-rate, short, and variable packets in delay sensitive applications. ATM's VBR-rt services enable statistical multiplexing for the higher layer requirements demanded by voice applications, such as compression, silence detection/suppression, and idle channel removal. Additionally, in contrast to AAL1 (which has a fixed payload), AAL2 offers a variable payload within cells and across cells.

Compression and signaling software, such as Mariner Networks' compression and signaling software, terminates the local signaling channels and provides inter-IAD proxy signaling over AAL5. This signaling provides for compressed calls that includes Robbed Bit/CAS modes, and out-of-band Common Channel Signaling (CCS) for a number of message oriented signaling protocols.

The IAD support compressed calls with in-band signaling (Robbed Bit/CAS) for non-ISDN T1/E1 interfaces and the following CCS variants when IAD is configured for ISDN PRI mode:

1. PRI Net5 User Mode
2. PRI Net5 Network Mode
3 PRI QSIG.

Figure 27 illustrates some of the network interconnection scenarios that can be implemented using a network of IADs and voice compression and multiplexing technologies.

Figure 27 has the following key attributes:

1. Any combination of AAL1 uncompressed and AAL2 compressed calls can be configured and carried by the IAD.
2. In addition to an AAL2 VCC between a pair of IADs, an AAL5 signaling VCC is required to carry the IAD's signaling protocol for switched, compressed voice/fax calls, such as Mariner Networks' proprietary signaling protocol for switched, compressed voice/fax calls.
3. Inter-IAD AAL2 compressed VCCs can be used to connect dissimilar PBX technologies (e.g., ISDN PRI using CCS to standard T1 using robbed bit signaling).
4. The IAD can also support analog interfaces that directly interface to fax machines, emulating the functions of a PBX to the attached devices.

Protocols and Standards Compliance

The IAD implements a combination of both standards-based and non-standards-based software protocols. The following sections provide an overview of these protocols.

AAL1 Protocol

The IAD implements Nx64K structured mode CES over AAL1, as defined in af-vtoa-0078.000. The IAD is loaded with conventional software configurable, on a per-VCC basis, to run either Basic or CAS-mode CES for configured trunk channels. Trunk channels carried via CES are transported in uncompressed, 64K PCM format. The IAD does not implement unstructured mode CES (as defined in af-vtoa-0078.000), nor does it implement SRTS clock recovery as defined for AAL1 transport by the ATM Forum and ITU.

AAL2 Protocol

The IAD implements a software based AAL2 implementation that is proprietary. This implementation utilizes the "general framework and Common Part Sublayer (CPS)" of the AAL

type 2 defined in ITU-T Recommendation I.363.2. The associated cell payloads comprise compressed voice/fax data output by the IAD compression engine.

It is preferred to implement standards-based software solutions wherever possible to maximize interoperability opportunities. Once the standards for AAL2 signaling have been agreed and accepted, such solutions will preferably be implemented into IAD's AAL2 voice processing software.

AAL5 Protocol

The IAD implements the ITU-T I.363.5-compliant AAL5 UBR transport mechanisms widely deployed today. This service is used to convey IAD voice signaling messages in conjunction with AAL2-based voice traffic.

Voice Compression

Voice compression is performed by IAD's compression engine that consists of software logic and a number of Digital Signaling Processors (DSPs). The IAD can be configured to operate with a number, such as 4 DSPs. Each DSP can support the processing of numerous, such as 8, voice channels concurrently. The IAD can be configured to support any set of the following voice encoding techniques:

1. G.711 PCM, 64Kbps
2. G.726 ADPCM, rates 16, 24, 32, and 40Kbps
3. G.727 EADPCM, rates 16, 24, 32, and 40Kbps
4. G.729A CS-ACELP and G.729B CS-ACELP, 8kbps rate
5. G.723.1A, rates 5.3 and 6.3Kbps.

Proprietary Protocols

As the ATM Forum and/or the ITU do not yet standardize signaling for AAL2, IAD's utilize the proprietary Helium™ signaling protocol to establish and tear down individual compressed voice calls. These calls are signaled using Robbed Bit/CAS/CCS modes on the

facility side, and converted to/from the IAD's proprietary "Q.931-like" signaling protocol for managing inter-IAD call states. Conventional signaling protocol may be used.

PBX Interface Mode

The IAD can operate in one of three modes: North American T1, Standard E1, and E1-based ETSI ISDN PRI.

In T1 mode, narrowband signaling is via the AB bit transitions in robbed bit frames of the T1 Super Frame (SF) or Extended Super Frame (ESF) multiframe. In E1 (non PRI) mode, narrowband signaling is via CAS AB bit transitions in slot 16 of all frames in the E1 (FAS/CAS or FAS/CAS-CRC4) multiframe. In E1 PRI mode, narrowband signaling is configurable as QSIG, PRI NET5 User Side, or PRI NET5 Switch Side, via CCS in timeslot 16 of all frames in the E1 (FAS/CAS or FAS/CAS-CRC4) multiframe.

Trunk Channel Signaling

IAD supports the following narrowband signaling protocols for trunk channel signaling. For each channel, one of the following may be selected as the signaling protocol:

1. Foreign Exchange Station Loop Start or Ground Start
2. Foreign Exchange Office Loop Start or Ground Start
3. E&M Immediate Start
4. E&M Delay Start
5. E&M Wink Start.

This operation is unavailable when the IAD is operating in PRI (Primary Rate Interface) mode.

Voice Coding Profiles

PCM (Pulse Code Modulation) voice samples from the PBX (Private Branch Exchange) interface are switched through the IAD's on-board Digital Signaling Processors (DSPs), on a per-call basis, in order to perform the required compression, silence suppression, voice activity

detection, and echo cancellation processes. All DSPs (up to a maximum of 4) are loaded with the same image at power up, which supports the following protocols (on a per channel basis, 8 channels per DSP):

1. G.711
2. G.729A and B
3. G.726
4. G.727
5. Standard Fax relay.

Configuration of the DSP feature set is achieved through the creation of "Voice Coding Profiles". A coding profile is a set of configuration parameters that is assigned to a compressed call. The information in the coding profile informs the DSP how to process and route the compressed call through the system.

Coding profiles with common characteristics must be configured on both IAD peers in order for a call to be successfully placed between them. At the originating end, a coding profile is assigned to a destination telephone number. When a call request for a particular destination is received from the telephony interface at the originating end, the parameters from the associated coding profile are negotiated with the remote peer via the IAD's proprietary signaling message elements. At the remote end, a coding profile will have been associated with the telephony destination through prior configuration.

Common elements from the originating side's coding profile and the destination side's coding profile are then negotiated and converged upon (via signaling) to create the set of parameters used to configure the associated DSP voice channels at both ends. Once this process is completed, the voice call is considered active.

Dial Plan Configuration

In addition to physical resource configuration (PBX mode, FXO, FXS, etc.), a dial plan that specifies how to route calls between IAD peers is required. The IAD maintains its own dial plan that contains the following information:

1. Dialed digit timeouts and termination sequences,
2. Narrowband hunt group definitions,
3. Broadband hunt group definitions, and
4. Forwarding criteria.

SNMP (Sample Network Management Protocol)

Standard MIB (Management Information Base) support for the IAD includes:

1. RFC 1406 Standard T1/E1 MIB, and
2. Supplemental MIB supporting ANSI T1.231.

Additionally, IAD is configured with its Enterprise MIB structure to facilitate the reporting of non-standard object elements such as ISDN PRI information.

Network Management Processing

This subsystem provides the facility to control and configure the IAD's different subsystems.

Overview

The Network Management Subsystem comprises four main components that enable a network operator to configure, control, report, and perform diagnostics upon the IAD. These elements are:

1. Configuration Management,
2. Connection Management,
3. Fault Management, and
4. Performance Management.

Configuration Management

This component provides functions to configure all aspects of the IAD's physical interfaces, signaling protocol parameters, and call control parameters. From a management perspective, this involves the following entities:

1. General node configuration,
2. E1/T1 port and subchannels,
3. BRI-ISDN, 10BaseT, V.35 , and RS-232C ports,
4. ATM and IMA ports,
5. Narrowband signaling,
6. Inter-IAD communications,
7. Voice coding profiles,
8. Routing, narrowband, and broadband addressing tables,
9. OAM segmentation end points table,
10. Frame Relay and IP interworking tables, and
11. CES configuration.

Connection Management

Connection Management is a set of functions that is used to track the various call or connection oriented entities and configuration of PVCs, including applications they support. From a node management perspective, this involves describing the details of:

1. Active call connections between narrowband and broadband resources,
2. Active broadband connections for the total system,
3. PVCs created for the broadband entities,
4. PVCs created for the narrowband entities, and
5. Call history information.

Fault Management

Fault Management is a set of functions that enable the detection, isolation, and correction of abnormal operation of the telecommunications parts of the network and its environment. From a node perspective, this tracks the following entities:

1. Physical facility and port failures,
2. Call control failures,
3. ATM OAM cell loopback tests, and
4. Sundry fault management and vendor-specific diagnostics.

Performance Management

Performance Management provides functions to evaluate and report upon the behavior of telecommunication/data equipment and the effectiveness of the overall network or network element. From a node management perspective, this involves general performance, traffic, and data collection routines against the following entities:

1. Physical layer performance monitoring of all ports,
2. Cell level performance monitoring, and
3. ATM layer protocol and performance monitoring.

Standards Compliance

The standards and compliance specifications relevant to IAD are.

ANSI Documents

1. T1.CBR-199X Draft – Broadband ISDN – ATM Adaptation Layer for Constant Bit Rate Services, Functionality and Specification, November 1992.
2. T1.102-1993, Digital Hierarchy, Electrical Interfaces, December 1993.
3. T1.107-1995, Digital Hierarchy, Formats Specifications, 1995.
4. T1.231-1993, Digital Hierarchy, Layer 1 In-Service Digital Transmission Performance Monitoring, September 1993.
5. T1.403-1995, Carrier-to-Customer Installation, DS1 Metallic Interface, March 1995.
6. T1.408-1990, Integrated Services Digital Network (ISDN) Primary Rate -- Customer Installation Metallic Interfaces Layer 1 Specification, September 1990.
7. T1.606, T1.606a, T1.606b Frame Relay Bearer Service, Architectural Framework and Service Description, ANSI, 1990.
8. T1.646-1995, Broadband ISDN, Physical Layer Specifications for User-Network Interfaces Including DS1/ATM, 1995.
9. EIA/T1A-547, Network Channel Terminal Equipment for DS1 Service, March 1989.

ATM/Frame Relay Forum Documents

11 AF-VTOA-0078.000, Circuit Emulation Service Interoperability Specification, Version 2.0, January 1997.

12. The ATM Forum, af-vtoa-0089.000, "Voice and Telephony Over ATM – ATM Trunking using AAL1 for Narrowband Services Version 1.0", July 1997.

13. The ATM Forum, af-phy-0086.000, "Inverse Multiplexing for ATM (IMA) Specification, Version 1.0, July 1997.

14. The ATM Forum, af-vtoa-0113.000, "ATM Trunking using AAL2 for Narrowband Services", Version 1.0, February 1999.

15. UTOPIA, An ATM-PHY Interface Specification, Level 2, Version 0.95, June 1995.

ATM User-Network-Interface Specification, Version 3.1, September 1994, ATM Forum.

16. UTOPIA, An ATM-PHY Interface Specification, Level 2, Version 0.95, June 1995, ATM Forum.

17. Network Working Group, RFC 1483, "Multiprotocol Encapsulation over ATM Adaptation Layer 5".

18. FRF.1.1, User-to-Network Implementation Agreement.

19. FRF.3.1, Frame Relay Forum Multiprotocol Over Frame Relay.

20. Frame Relay/ATM PVC Network Interworking Implementation Agreement, Document Number FRF.5, December 20, 1994.

21. Frame Relay Forum. Frame Relay/ATM PVC Service Interworking Implementation Agreement, Document Number FRF.8, April 15, 1995.

IETF

22. RFC 1483 Multiprotocol Encapsulation Over AAL5, July 1993.

23. RFC 1490 Multiprotocol Interconnect Over Frame Relay, July 1993.

24. RFC1577 Classical IP and ARP over ATM, January 1994.

ITU Documents

25. ITU-T Recommendation G.168, Digital Network Echo Cancellers, April 1997.

26. Draft new ITU-T Recommendation I.363.2, B-ISDN ATM Adaptation Layer Type 2 Specification, February 1997.

27. ITU-T Recommendation I.362 B-ISDN ATM Adaptation Layer(AAL) Functional Description.

28. ITU-T Recommendation I.363 B-ISDN ATM Adaptation Layer(AAL) Description.

29. Recommendation G.703, Physical/Electrical Characteristics of Hierarchical Digital Interfaces, 1991.

30. Recommendation G.704, Synchronous Frame Structures Used at Primary and Secondary Hierarchical Levels, 1991.

31. Recommendation G.706, Frame Alignment and Cyclic Redundancy Check (CRC) Procedures Relating to Basic Frame Structures Defined in

32. Recommendation G.704, 1991.

33. Recommendation G.804, ATM Cell Mapping into Plesiochronous Digital Hierarchy (PDH), January 1993.

34. Recommendation G.823, The Control of Jitter and Wander Within Digital Networks Which are Based on the 2048 kbit/s Hierarchy, 1993.

. Recommendation G.826, Error Performance Parameters and Objectives for International, Constant Bit Rate Digital Paths at or above the Primary Rate, 1993.

35. Recommendation G.832, Transport of SDH Elements on PDH Networks: Frame and Multiplexing Structures, 1993.

36. Recommendation I.233.1, Framework for providing additional packet mode bearer services, ITU-T, 1988.

37. Recommendation I.370, Congestion management for the ISDN Frame Relaying bearer service, ITU-T, 1988.

38. Recommendation I.431, Integrated Services Digital Network (ISDN) User-Network Interface, Primary Rate UNI Layer 1 Specification, March 1993.

39. Recommendation I.432, Broadband Integrated Services Digital Network (B-ISDN) User-Network Interface, Physical Layer Specification, March 1993.

40. Recommendation I.610, Broadband Integrated Services Digital Network (B-ISDN) Operation and Maintenance, Principles and Functions, March 1993.

41. Recommendation Q.922 ISDN Data Link Layer Specification for Frame Mode Bearer Services, 1992.

42. ITU-T Recommendation Q.931, DSS1 – ISDN User-Network interface layer 3 specifications for basic call control.

43. Recommendation Q.933, Digital Subscriber Signaling System No. (DSS 1), Signaling For Frame Mode Basic Call Control, ITU-T, 1993.

<u>Other Related Documents</u>

44. EN50082-1 "Electromagnetic compatibility, Generic immunity standard, Part 1: Residential, commercial and light industry". EN 50082-1:1997 (or BS EN 50082-1:1998).

45. ENV 50204 "Radiated electromagnetic field from digital radio telephones - Immunity test". ENV 50204:1995.

46. IEC 61000-4-2 "Electromagnetic compatibility (EMC), Part 4-2: Testing and measurement techniques, Electrostatic discharge immunity test". IEC 61000-4-2 Consol. Ed. 1.1 (incl. am1), 1999-05.

47. IEC 61000-4-3 "Electromagnetic compatibility (EMC), Part 4-3: Testing and measurement techniques, Radiated, radio-frequency, electromagnetic field immunity test". IEC 61000-4-3 - Consol. Ed. 1.1 (incl. am1), 1998-11.

48. IEC 61000-4-4 "Electromagnetic compatibility (EMC), Part 4: Testing and measurement techniques, Section 4: Electrical fast transient/burst immunity test. Basic EMC Publication". IEC 61000-4-4 - Ed. 1.0, 1995-01.

49. IEC 61000-4-5 "Electromagnetic compatibility (EMC), Part 4: Testing and measurement techniques, Section 5: Surge immunity test". IEC 61000-4-5 - Ed. 1.0, 1995-02.

50. IEC 61000-4-6 "Electromagnetic compatibility (EMC), Part 4: Testing and measurement techniques, Section 6: Immunity to conducted disturbances, induced by radio-frequency fields". IEC 61000-4-6 - Ed. 1.0, 1996-04.

51. IEC 61000-4-8 "Electromagnetic compatibility (EMC), Part 4: Testing and measurement techniques, Section 8: Power frequency magnetic field immunity test. Basic EMC Publication". IEC 61000-4-8 - Ed. 1.0, 1993-06.

52. IEC 61000-4-11 "Electromagnetic compatibility (EMC), Part 4: Testing and measuring techniques, Section 11: Voltage dips, short interruptions and voltage variations immunity tests". IEC 61000-4-11 - Ed. 1.0, 1994-06.

53. EN50081-1 "Electromagnetic compatibility, Generic emission standard, Part 1: Residential, commercial and light industry". EN 50081-1:1992.

FCC Part 15 "RADIO FREQUENCY DEVICES". Downloaded October 1998. Federal Communications Commission, USA.

54. EN55022 "Information technology equipment, Radio disturbance characteristics, Limits and methods of measurement", CISPR 22 - Ed. 3.0 - Bilingual, 1997-11, or EN 55022:1998.

55. EN 55014-1 "Electromagnetic compatibility, Requirements for household appliances, electric tools and similar apparatus, Part 1: Emission, Product family standard". EN 55014-1:1993/A2:1999.

56. EN 61000-3-2 "Electromagnetic compatibility (EMC), Part 3-2: Limits - Limits for harmonic current emissions (equipment input current up to and including 16A per phase)". EN 61000-3-2:1995/A2:1998.

57. EN 61000-3-3 "Electromagnetic compatibility (EMC), Part 3: Limits - Section 3: Limitation of voltage fluctuations and flicker in low-voltage supply systems for equipment with rated current up to 16 A". EN 61000-3-3:1995.

58.. IEC60950 "Safety of information technology equipment." IEC 60950 (1999-04) (Ed.3).

59. EN60950 "Safety of information technology equipment." EN 60950:1992/A4:1997.

60. UL 1950 "Standard For Safety For Information Technology Equipment", UL 1950_3 third edition 1995.

61. UL 1459 "Standard For Safety For Telephone Equipment", UL 1459 third edition 1995.

62. EN 41003 "Particular safety requirements for equipment to be connected to telecommunication networks", EN 41003:1998.

Books

63. Demystifying ATM/ADSL; Busby, Michael; Wordware Publishing, Inc.; 1998.

64. QoS & Traffic Management in IP & ATM Networks; McDysan, David; McGraw-Hill; 2000.

65. ATM Theory and Application; McDyson, David E. and Spohn, Darren L.; McGraw-Hill; 1998.

66. ATM for Dummies; Gadeck; Cathy and Heckart, Christine; IDG Books Worldwide, Inc.; 1997.

67. Networking for Dummies; Lowe, Doug; IDG Books Worldwide, Inc.; 1994.

The above documents and books are incorporated by reference herein.

Related websites include: www.atmforum.com; www.cis.ohio-state.edu/~jain/refs/atm-book.htm (extensive list of ATM network related books); www.networking.ibs.com/atm/atmover.html; //members.tripod.com/~vbkurnar/atm.html (extensive lists of glossaries, acronyms, telecommunications associations, organizations and forums); www.marinernetworks.coml; www.dexteraccess.com.

BRIEF DESCRIPTION OF THE DRAWINGS

Figure 1 is a simplified, partly schematic perspective view of an Integrated Access Device For Asynchronous Transfer Mode (ATM) communications Interface Module according to the present invention.

Figure 2 is a top-level block diagram of the device of Figure 1, showing major components thereof.

Figure 3 is a more detached block diagram of the device of Figure 1.

Figure 4 is a schematic diagram showing software modules of the device of Figure 1.

Figure 5 is a block diagram of a voice card module according to the present invention useable with the device of Figure 1.

Figure 6 is a block diagram of another embodiment of a voice card module according to the present invention and useable with the device of Figure 1.

Figure 7 is a perspective view of the device of Figure 1, showing three different modules according to the present invention plugged into three different expansion ports of the device.

Figure 8 is a schematic view showing the device of Figure 1 interfaced with various networks and devices through its expansion port modules.

Figure 9 is a perspective view of a T1/E1 IMA Interface Module according to the present invention and useable with the device of Figure 1, that Module adapted to perform inverse multiplexing of up to four T1/E1 data lines.

Figure 10 is a perspective view of a Synchronous Serial Interface Module according to the present invention and useable with the device of Figure 1, that module adapted to receive data in either an ATM cell or framed mode.

Figure 11 is a schematic view similar to that of Figure 8, but showing additional networks and devices interfaced with the device of Figure 8.

Figure 12 is a front panel view of an ATM/FR T1/E1 Interface Module according to the present invention.

Figure 13 is a front panel view of an ATM/FR T1/E1 IMA Interface Module according to the present invention.

Figure 14 is a front panel view of an ATM DS-3/E3 Interface Module according to the present invention.

Figure 15 is a front panel view of an ATM OC-3/STM-1 Interface Module according to the present invention.

Figure 16 is a front panel view of an ATM/FR SDSL Interface Module according to the present invention.

Figure 17 is a front panel view of an ATM/FR HDSL2 Interface Module according to the present invention.

Figure 18 is a front panel view of an FR V.35/X.21 Interface Module according to the present invention.

Figure 19 is a front panel view of Switched 10/100 Base T Interface Module according to the present invention.

Figure 20 is a front panel view of a PBX T1/E1 Interface Module according to the present invention.

Figure 21 is a front panel view of a PBX T1/E1/PRI + BRI Interface Module according to the present invention.

Figure 22 is a front panel view of a ISDN BRI Interface Module according to the present invention.

Figure 23 is a diagrammatic view showing Inverse Multiplexing (IMA) logic flow implemented in the device of Figure 1.

Figure 24 is a diagrammatic view showing network interworking mapping implemented in the device of Figure 1.

Figure 25 is a diagrammatic view showing service interworking mapping implemented by the device of Figure 1.

Figure 26 is a diagrammatic view showing the device of Figure 1 interfaced with various networks and PBXs to form CES-based voice connections.

Figure 27 is a view similar to that of Figure 25, but showing AAL-2 based voice connections.

Figure 28A is a simplified block diagram of an Application Specific Integrated Circuit (ASIC) module comprising part of the device of Figure 1, which is operably interconnected with other components of the device.

Figure 28B is a more detailed version of the block diagram of Figure 28A.

Figure 29 is a table showing contents of a bubble register associated with the ASIC of Figure 28.

Figure 30 is a diagram showing the structure of the register of Figure 29.

Figure 31 is a table showing the arrangement of port scheduling registers of the device of Figure 1.

Figure 32 is a flow chart showing port scheduling of the device of Figure 1.

Figure 33 is a table illustrating operation of the port scheduling portion of the bubble table of the device of Figure 1.

Figure 34 is a table illustrating operation of the scheduler table function of the device of Figure 1.

Figure 35 is a flow chart illustrating a Ci (Connection Index) activation process implemented by the device of Figure 1.

Figure 36 is a diagrammatic view of data structures of the device of Figure 1.

Figure 37 is a table indicating assignments of port numbers for the device of Figure 1.

Figure 38 is a group of 4 tables illustrating logical organization of the apparatus of Figure 1.

Figure 39 is a table showing FIFO sizes for the device of Figure 1.

Figure 40 is a table showing the organization of an IN STAT register for the device of Figure 1.

Figure 41 is a block diagram of a Cell Pointer block of the device of Figure 1.

Figure 42 is a block diagram of a Tdm Resolution block of the device of Figure 1.

Figure 43 is a block diagram showing a prior art scheduler for multiple qualities of service.

Figure 44 is a block diagram showing a single scheduler to fully service multiple qualities of service according to the present invention.

Figure 45 is a flow chart showing prior art multiple queues associated with a buffer pool.

Figure 46 is a flow chart showing a Beaded Buffer Pointer Chain With Intermediate Pointers according to the present invention.

Table 1 is a list of Interface Modules useable in the device of Figure 1.

DESCRIPTION OF THE PREFERRED EMBODIMENTS

1. **Overview**: (Product Specification MAKO-Dexter - 3000, pp. 1-6, Revised 1.0.0.)

2. 2-Page Dexter 3000 Integrated Access Device Data Sheet.

3. 1-page Dexter 3000 Interface Module, T1 and E1 IMA.

4. 1-page Dexter 3000 Interface Module, Synchronous Serial.

5. **Product Guide**

   a. Chapter 2. Introduction.

   b. Chapter 3. Features and components.

   c. Chapter 4. Functional description.

   d. Chapter 5. Standards compliance.

   e. 1-page index

In the description of the invention titled "Integrated Access Device For Asynchronous Transfer Mode ATM Communications" contained in this specification, the invention is sometimes referred to as a Dexter 3000 IAD (Integrated Access Device), or Dexter. The Integrated Access Device for ATM according to the present invention includes an integrated circuit module which comprises an array of logic gates and flip-flops which are interconnected to form a cell switching fabric. The cell switching fabric functions in cooperation with other components of the Integrated Access Device to segment and re-assemble cell queues, and includes a cell forwarding architecture that implements a cell scheduler function. This Integrated Circuit Module is preferably an Application Specific Integrated Circuit (ASIC) but may optionally be a Programmable Logic Array (PLA). In this specification, the integrated circuit which contains the cell switching fabric is referred to interchangeably as MAKO or eXpedite™ processor.

6. **Operation of the Invention.**

   (a) Product Specification MAKO

   (b) Scheduler High Level Information. Pp. 1-15.

   (c) Further Identified Aspects of the Invention.

      1. A Single Scheduler to Fully Service Multiple Qualities of Service.

2. Algorithm to Assign Scheduler Resources to Multiple Ports in Correct Proportions.

3. Beaded Buffer Pointer Chain With Intermediate Pointers.

4. Fractional Interval Times for Fine Granularity Bandwidth Allocation.

5. Multiple Preemptive CBR's for Precise Port Pacing Control.

6. Partitionable Page Shifter With Self-Timing Xor Chain.

Product Specification
Mako-Dexter-3000

Document number
Revision 1.0.0

57C

1. Introduction

1.1. Document Scope

This document describes the Mako-Dexter-3000, an Integrated Access Device (IAD) for Asynchronous Transfer Mode Communications according to the present invention, supporting data and voice in the customer premise. Mako-Dexter Device-3000 is a 1U high chassis based product. A modular design enables it to support several configurations.

1.2. Mako-Dexter-3000 Description

Mako-Dexter-3000 is a functional bridge and IP router incorporating Ethernet, Frame Relay, ATM and voice technologies. With ATM switching and scheduling at its core, Mako-Dexter-3000 fully supports Quality of Service in ATM and is able to impose ATM QoS onto its non-ATM ports. It supports ATM PVC 's and SVC's with UNI 3.0, 3.1 and 4.0 signaling. AAL-5 is supported for data. Mako-Dexter-3000 incorporates Frame Relay over ATM Interworking standards FRF.8 and FRF.5. AAL-1 and AAL-2 are supported for voice. Both digital or analog voice are supported. Figure 1 is is a general indication of the function, look and feel.

Figure 1: Mako-Dexter-3000

The Mako-Dexter-3000 is modularized as shown in figure 2. The Main Board performs the core ATM switching and scheduling functions and Frame Relay to ATM Interworking. The Voice Processor performs voice compression and conversion of TDM voice channels to AAL-1 or AAL-2. The other modules provide physical interfaces.

Figure 2: Major Components

The Main Board has three expansion ports that connect to IAD input/output modules.

The first expansion port is for WAN access. It connects to one of several possible daughter cards. One type of daughter card is a T1/E1 LIU. The T1/E1 LIU daughterboard provides up to 8 ports of Frame Relay over T1/E1. Mariner LIMO's are another type of daughter card that can be attached. Mariner LIMO's are current products that provide 4xIMA T1/E1, DS3/E3 and OC-3/STM-1 WAN interfaces. Other daughter cards include DSL and ISDN. DSL interfaces may be SDSL or ADSL.

The second port is for LAN access. It connects to a single or to a quad 10/100 Ethernet module.

The third port is for voice access. It connects to a Voice Processor module. This module, in turn, connects to either a digital voice interface module or an analog voice interface module. The digital voice module provides one channelized T1/E1 port. The analog voice module provides 12 analog ports.

2. Product Architecture

2.1. Main Board

The Mako-Dexter=3000 Main Board is architected as shown in figure 3.

Figure 3: Mako-Dexter-3000 Main Board Block Diagram

The Main Board contains an ARM-7 CPU, imbedded in the Helium, on which the ATMOS operating system is run. ATMOS contains packet bridging and IP routing, TFTP, DHCP, NAT, ATM signaling, encapsulation (RFC 1483), CIP (RFC 1577 and RFC 1755), LANE and SNMP 16 MB of SDRAM is available for ATMOS and application firmware. Boot and application code is stored in compressed format in a 2 MB Flash. The ARM-7, which runs internally at 48 MHz, interfaces to SDRAM, Flash and other peripherals via a 24 MHz local bus.

Data switching and routing on up to 4000 logical connections is performed in the Mako, a proprietary Mariner ASIC. Mako was initially implemented in an <u>Altera 20K400E FPGA</u> until the design became stable and economics dictated cost reduction into a gate array/standard cell ASIC. ATMOS performs programming and configuration of Mako through the local bus. Mako operates at 98 MHz and utilizes its own dedicated SDRAM. In the Mako-Dexter-3000 application, Mako has 6 data interfaces: 3 E2/T2 TDM interfaces and 3 Utopia level 1/2 interfaces. The TDM interfaces operate at 8.192 MHz in E1 mode and 6.176 MHz in T1 mode. The Utopia interfaces operate at 24 MHz. The two TDM interfaces are routed to the WAN Port when the WAN Port is connected to the T1/E1 LIU daughter card or to the ISDN daughter card. When the Wan Port is connected to a LIMO or a DSL daughter card, one of the Utopia ports is routed to the WAN Port. Another Utopia interface is dedicated to the Voice Port. The third Utopia interface is routed to the LAN Port.

The Dual Port RAM is a mailbox RAM used to communicate with the Voice Processor. For lower cost, the Dual Port RAM may be implemented as a priority arbitrated window into a block of SDRAM rather than via actual dual ported RAM.

The ID Interface allows both the ATMOS system and the IBM Switch to read and write the 1024 bit ID EEPROM. The ID EEPROM is programmed during manufacturing functional test with a Serial Number and other configuration information.

A Serial port is also available. This port is multiplexed under software control serve as CLI console for either the Main Board or the Voice Processor. A Telnet session will also serve as an alternate console.

The Helium contains a 10Base-T MAC and transceiver. The signals from this Helium block, including LED's are passed from the Helium through the LAN connector to the LAN board.

Not shown in the diagram is glue logic, implemented in FPGA or other convenient technology. This glue logic performs device decodes, buffering and other details. It also contains a register that can be written by the Helium which controls 8 LED's. The drivers for these LED's are terminated on an 8 pair .100" center header. A separate cable assembly can be used to connect these signals to a similar 8 pair header on the LIMO and LIU I/O Extension card. The I/O extension card is the small PC board that passes data signals from the high density connectors of the LIMO and LIU to the individual RJ48C connectors on the front panel of the Dexter 3000.

Figure 4 shows the software structure for the Main Board.

Figure 4: Main Board Software Structure

The ATMOS kernel is the real time operating system core that manages and schedules software operation. With the exception of 10Base-T Ethernet data to and from the internal 10Base-T controller of the Helium, all network traffic is forwarded through and by the Mako hardware. Although not a software block, the Mako bridging and routing function is shown in the diagram to show its functional relationship to the functions of the software blocks.

Network data encountered by the Mako hardware that is destined for either the internal node or the Helium 10Base-T port is passed by the Mako hardware to the LocPt Rcv module in the form of ATM cells. OAM cells are processed and appropriate response cells passed to the LocPt Send module which passes them to the Mako hardware for forwarding to the network.

LocPt Rcv cells that are destined to go out on the Helium 10Base-T port are passed to the Helium Bridging module. The Helium Bridging module passes them to the Helium SAR which converts them to packets. Then they are sent out on the Helium 10Base-T port.

LocPt Rcv cells that are destined for the internal node are converted by the Helium SAR into packets and sent to the FR LMI, FR Signaling, ATM ILMI, ATM Signaling or IP modules as appropriate.

Packets coming in from the Helium's 10Base-T port are examined to see if they are destined for the internal node. If for the internal node, they are forwarded to the IP module, else they are converted to cells by the Helium SAR and presented to the Helium Bridging module which passes them to the LocPt Send module which passes them to the Mako hardware which routes them to the appropriate network port.

The IP module forwards packets upward for UDP, TCP or ICMP processing or forwarding to upper layers in the traditional manner of IP protocol stacks. Packets reaching Telnet are converted to command strings and passed to the CLI for parsing. The CLI can also receive command strings from the Serial Port Driver. The CLI passes parsed commands to the Control & Configuration module. Packets reaching the SMNP are also parsed and the resulting commands passed to the Control and Configuration module. The Control & Configuration module processes command, performing whatever action is appropriate, and returns response information the whichever module, i.e. CLI or SMNP, passed the command to it. CLI and SNMP, in turn, encode the responses appropriately into packets and pass the packets back down. Downward passing continues until the responses are sent out to the appropriate network connection.

2.2. Voice Processor

The voice processor has gone through two iterations. The first was quicker to market but more costly than the second. The first retains the architecture of the current Dexter 2200, including the MPC860 processor with its own operating system and protocol stacks and is called the "860VoiceCard" in this document. The second replaces the MPC860 and its supporting peripherals by an FPGA. The FPGA provides polling of the DSP's and routing of DSP data through otherwise unused time slots on the TDM highway. This second iteration is called the "MakoVoiceCard" in this document. There may be more than one version of each card, with differing external connection options.

Figure 5 shows the architecture of the 860VoiceCard with digital PBX option. A single T1/E1/ISDN PRI connection to a PBX will be supported. A similar diagram could be imagined with analog POTs connections. Up to 4 analog connections will be supported.

Figure 5: 860VoiceCard Block Diagram

This 860VoiceCard contains an MPC860 CPU on which the pSOS operating system runs. The MPC860 also contains a TDM interface, internal SAR and Utopia interface. 16 MB of SDRAM is available for pSOS and voice processing software adapted from Telenetworks, Telogy, Inverness and Ficon. These are loaded from Flash where they are stored in compressed format. A Serial Port serves as the CLI console. The serial port is multiplexed on the Main Board. A hot key combination will toggle the serial port between the Main Board and the Voice Processor. Alternatively, a Telnet session can serve as console. It too will toggle between the Main Board and the Voice Processor on the same hot key combination. A jumper selection on the main board will allow the console to be forced to either one or the other system.

The Dual Port RAM provides a mailbox interface between the Voice Processor and the Main Board.

Uncompressed incoming voice data goes across the TDM highway to the MPC860 where it is converted to an AAL-1cell stream by the Inverness software before being forwarded to the Main Board via the Utopia interface. Incoming voice data streams that are to be compressed or otherwise procesed are stripped off the TDM highway by Telogy software in the DSP's. Processed data is read by the MPC860 from the DSP parallel bus and is then converted by the Inverness software into AAL-2 before being forwarded via the Utopia interface. Incoming ATM streams are converted in the MPC860 to voice data. Voice channels may be routed through DSP's for decompression and other signal processing before being sent out on the TDM highway to the LIU.

Figure 6 shows the architecture of the MakoVoiceCard with digital PBX option. Up to two T1/E1/ISDN PRI connections to a PBX will be supported. A similar diagram could be imagined with analog POTs connections. Up to 16 analog connections will be supported.

Figure 6: MakoVoiceCard Block Diagram

Incoming voice data is put into TDM highway time slots by the LIU's. The TDM highway has twice as many time slots as the LIU's can occupy. The Mako on the Main Board can read the unprocessed data from the those time slots on the TDM highway which passes through the Voice Port. The Mako can then convert the data to AAL-1 or AAL-5 ATM cell streams for CES or packetized voice and forward it to the network. Alternatively, the Main Board processor can program the DSP's to read voice streams from the incoming time slots for processing and compression. The FPGA polls the DSP's and extracts their processed data from the parallel bus, then inserts it into otherwise unused time slots. The Mako on the Main Board can read processed data from these time slots, package it into AAL-2 cell streams and forward it to the network.

2.3. WAN Interface

As previously noted, a number of WAN Interface cards can be connected, including the LIU used with Fraim-IBM, Mariner LIMO's and future ISDN and DSL modules.

2.4. LAN Interface

Two configurations of Ethernet interfaces are provided. Each contains a Packet Processor ASIC that performs a table lookup translation between Ethernet Mac addresses and ATM VPI/VCI combinations and perform RFC 1483 and RFC 1577 encapsulation over AAL-5 conversion. One configuration supports a single 10/100 Ethenet connection. The other switches between 4 local ports and the Main Board.

2.4. LAN Interface

Two configurations of Ethernet interfaces are provided. Each contains a Packet Processor ASIC that performs a table lookup translation between Ethernet Mac addresses and ATM VPI/VCI combinations and perform RFC 1483 and RFC 1577 encapsulation over AAL-5 conversion. One configuration supports a single 10/100 Ethernet connection. The other switches between 4 local ports and the Main Board.

Dexter® 3000

Integrated Access Device

The Mariner Networks Dexter® 3000 Integrated Access Device (IAD) offers a fully scalable, easily affordable, low-speed, branch-office-access solution that enables companies to extend their ATM core network to all remote locations. It enables access to the broadband ATM backbone from existing legacy equipment without costly upgrades.

Multiservice Convergence Applications and Customers

- Healthcare – Telemedicine
- Education – Distance Learning
- Finance
- Retail
- International, National, and Regional Service Providers
- ILECs, CLECs, ISPs, ICPs
- Military Command/Field Communications
- Local, Regional/State, National Government
- Corporate offices
- Utilities

Overview

The Dexter product family is a series of modular integrated access devices (IADs) that enable small and medium-sized offices to connect to integrated broadband services. The 3000 platform is the most high-performance, cost-effective, multiservice access platform available for connecting all of the office's voice, data, and video networking equipment to low-speed, scalable, wide-area network services. Its modular, scalable design allows the customer to specify only those features required to meet current needs but maintain flexibility for future growth.

Bandwidth use on low-speed access links must be carefully managed to maximize the utility of this scarce resource. Based on Mariner Networks' proprietary **eXpedite**™ technology, the Dexter 3000 uses fine-grained ATM scheduling to get the most out of this finite resource.

The **eXpedite** architecture allows network managers to prioritize traffic flows across their WAN. For example, e-mail flows will not interfere with voice traffic, just as real-time transaction data will not be delayed by Web surfers.

The Dexter 3000 can also grow with the needs of the business by scaling up to four T1s or E1s of inverse-multiplexed ATM, while maintaining wire-speed bridging, routing, and interworking.

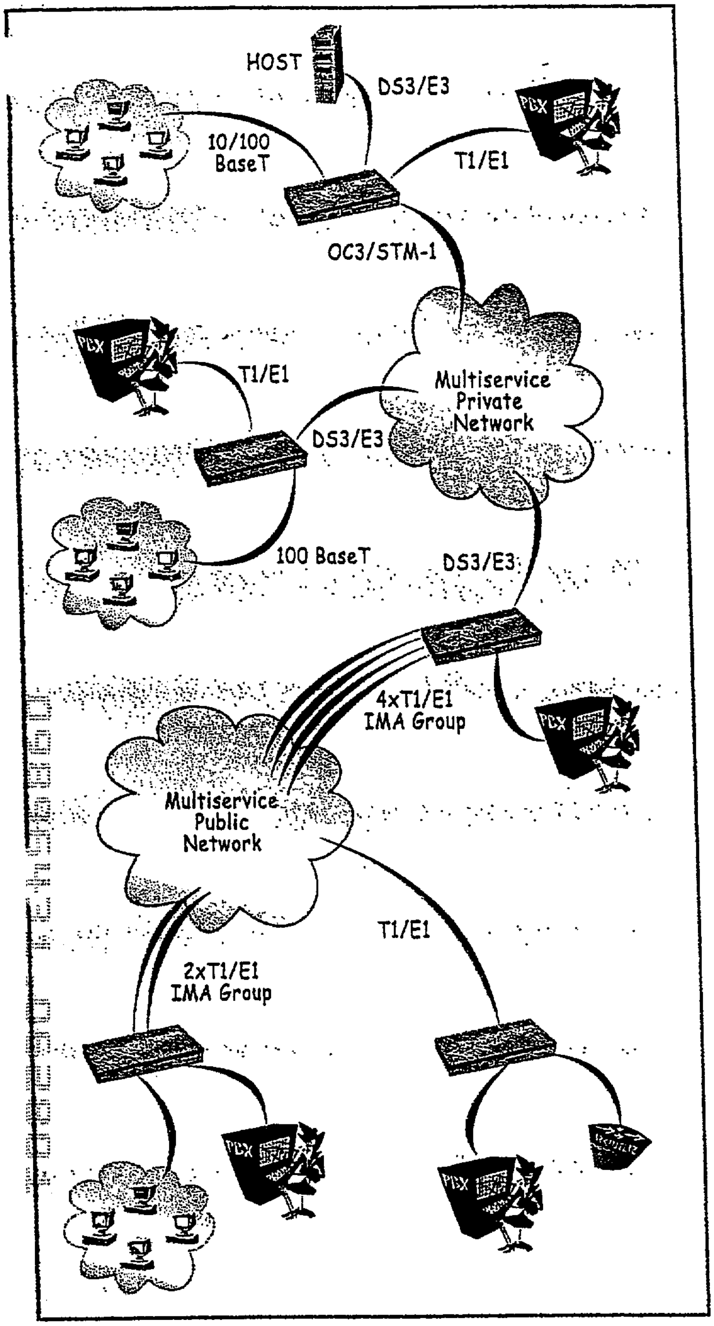
The Dexter family of multiservice IADs offer high-quality, low-bandwidth, branch and regional office access solutions designed to address the needs of the small office through to the central headquarters. Through Dexter's highly versatile modular approach, every access solution requiring the efficient convergence of voice, data, and video into a single managed communications link can be accommodated cost-effectively and simply.

Key Benefits

- Reduces equipment and line rental costs through the consolidation of separate voice, data, and video circuits onto a single cost-effective, low-speed ATM path.
- Reduces monthly access service costs by maximizing bandwidth efficiency through intelligent, fine-grain control over traffic flows.
- Improves small- to medium-sized office productivity by enabling corporate-centric applications, such as video distance learning or company board-level broadcasts, to be accessed easily and economically.
- Enhances interoffice communications by allowing offices and branches to become an intrinsic part of the corporate VPN without the need for costly access switches and high-speed communication lines.
- Simplifies network management monitoring and reporting processes through reduced communications equipment and a single networking topology.
- Provides an economical bandwidth growth strategy by applying ATM inverse multiplexing technology while maintaining wire-speed performance.

Dexter® 3000
Integrated Access Device

REAL T1/E1 Multiservice Access!



The Dexter 3000's performance, economy, and versatility allow network operators to deploy multiservice access in a broad array of applications.

Module-Enabled Features

- Integrated multiservice access device that provides PBX voice, Frame Relay, Ethernet, T1/E1, IMA, multiple ATM and xDSL interfaces, and BRI video and data services
- Frame Relay to ATM interworking
- AAL2 voice multiplexing with VBR-rt QoS
- Industry compliant silence suppression, comfort noise insertion, compression, and echo cancellation techniques
- Circuit Emulation Services via AAL1
- Wire-speed IP bridging and routing
- Local and remote management via Dexter's Messenger™ SNMP application
- Compliant with industry-standard protocols and interfaces

Technical Description

- 3 -slot chassis – any module in any slot
- Ethernet 10BaseT port
- RS-232 DB9 management port
- Dexter eXpedite™ processor providing:
  - Layer 3 wire-speed switching
  - Frame Relay to ATM interworking (FRF.5 and FRF.8)
  - Frame Relay LMI to T1.617 annex D and Q.933 annex A
  - Spanning Tree
  - RFC 1483 bridged and routed IP
  - NAT and DHCP server functions
  - Static routing via RIPv1 and v2
  - ATM UNI 3.0, 3.1, and 4.0
  - ATM AAL5 adaptation
  - ATM PVC and SVC
  - ATM scheduler for end to end QoS
  - PPP over ATM
  - LANE Emulation Server v1.0
  - Telnet access
  - Configuration protected via password
  - SNMP Version 1, MIB II

Mechanical

Size: 19" W x 12" D x 1.75 H (1U)
Weight: 6.5lbs
Mounting configurations:
- Desktop, rack, wall-mount
- 90-265 VAC 50/60Hz
- Max power: 40W
- 0° - 50° C operating temperature
- Humidity: 5% - 90%, non-condensing

MARINER NETWORKS
An Odetics Company

The Company

Mariner Networks, Inc., a wholly owned subsidiary of Odetics, Inc., manufactures components and complete solutions for the ATM Wide Area Networking communities and branch-office-access applications. The company's products include ATM subsystems, Frame Relay to ATM Interworking and ATM access concentrators for handling voice, data, and video traffic. Mariner supplies equipment to many OEMs and end users through offices located in the US, Europe, and Asia.

1585 South Manchester Avenue
Anaheim, CA 92802-2907
Tel 714-780-7685
Toll free 888-329-4487
Sales 714-780-7987
Fax 714-780-7696
E-mail sales@marinernetworks.com

Odetics Europe
Tel +44(0) 118 927 4607
Fax +44(0) 118 944 0069
E-mail ixb@odetics.co.uk

www.marinernetworks.com
www.dexteraccess.com

64

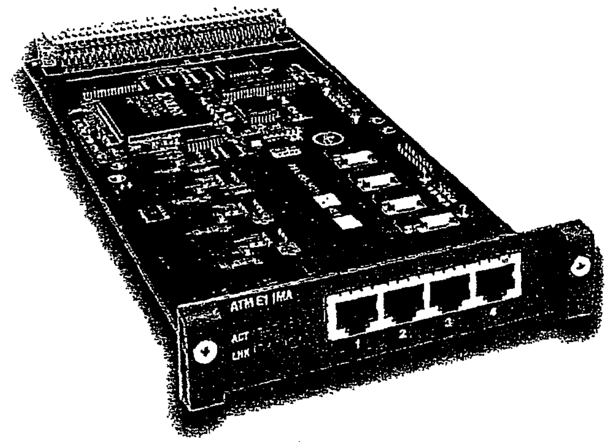
C

Dexter® 3000 Interface Module

T1 and E1 IMA

Key Features

- Module operates in Inverse Multiplexing over ATM (IMA) mode. Supports one or two logical IMA groups. Single group can consist of up to 4 ports (up to 8Mbps in a single ATM VC in E1 mode)
- Operates in either T1 (1.544Mbps) or E1 (2.048Mbps)
- Ungrouped links operate as standard T1s or E1s
- Balanced 100/120 Ohm physical interface
- ATMF UNI 3.0/3.1
- Integrated CSU/DSU functionality
- Multiple modules may be inserted into a single Dexter® chassis



Introduction

The T1/E1 IMA Interface Module provides connectivity of up to 4 x 2Mbps circuits into a single logical IMA group thus allowing ATM PVCs or SVCs to utilize all available bandwidth within the group. The module enables connectivity between the Dexter® family of Integrated Access Devices (IADs) or other ATM IMA compliant vendor equipment.

When used in the Dexter 3000 platform, the user can access the powerful functionality of the **eXpedite**™ architecture: converged voice, video, and data wide-area network access; wire speed protocol interworking, IP routing and bridging; ATM quality of service for all data flows; and much more.

Key Benefits

- IMA technology enables scalable T1 or E1 bandwidth on demand without needing to jump to DS-3 or E3 circuits
- Enables organizations to maximize available bandwidth as business volumes grow
- Allows carriers to maximize utility of existing copper loop assets
- Lowers the cost of ownership through by optimizing network resources efficiently and simply

Specifications

CSU/DSU Function

• Connector:	RJ48C
• Bit rate:	1.544Mbps or 2.048Mbps
• Line Coding:	B8ZS or HDB3
• Clock source:	Internal or external
• Impedance:	100/120Ohms
• Framing:	D4 (SF) or ESF, FAS, CRC4
• Line Build Out:	0-133ft, 133-266ft, 266-399ft, 399-533ft, 533-655ft, 0db, -7.5db, -15db,-22.5db

IMA Function

- Supports 2 logical IMA groups
- Passthrough mode for ungrouped links
- Performance and status reporting
- Automatic detection and recovery from circuit failure

Standards

- ATMF: UNI 3.1, AF-PHY-0016, AF-PHY-0064, and AF-PHY 00086.000d
- ANSI: T1.102, T1.107, T1.231, T1.403, T1.408, T1.646, EIA/TIA-547
- ITU-T: G.703, G.704, G.706, G.804, G.823, G.826, G.832, I.431, I.432, I.610
- ETSI: ETS 300 166, ETS 300 167, ETS 300 213, ETS 300 247, ETS 300 248, ETS 300 299, ETS 300 337, ETS 300 418, ETS 300 419, ETS 300 420

MARINER NETWORKS
An Odetics Company

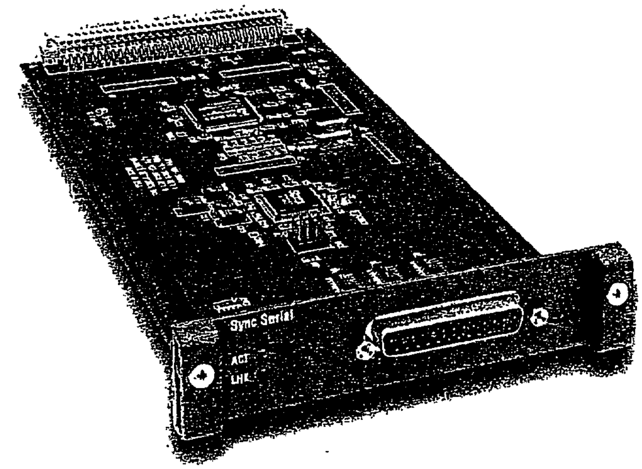
1585 South Manchester Avenue, Anaheim, CA 92802-2907
Tel 714-780-7685 Toll free 888-329-4487 Fax 714-780-7696
Sales 714-780-7987 E-mail sales@marinernetworks.com
Odetics Europe
Tel +44(0) 118 927 4607 E-mail ixb@odetics.co.uk Fax +44(0) 118 944 0069
www.marinernetworks.com www.dexteraccess.com

65

Dexter® 3000
Interface Module
Synchronous Serial

Key Features

- Single port, up to 8Mbps full duplex bit rate
- Operates in either ATM cell or Framed mode
- Frame Relay service interface supporting FRF.5 and FRF.8 FR/ATM interworking
- DB25 physical interface
- Cable adapter options are available that provide common physical interface connectors such as V.35, X.21, RS-449, and RS-530
- Multiple modules may be inserted into a single Dexter® chassis



Introduction

The Synchronous Serial Interface Module provides connectivity between the Dexter® family of Integrated Access Devices (IADs) and other ATM or Frame Relay compliant devices via a variety of cable connection interfaces. This module can be used to support legacy equipment such as Frame Relay routers and other framed mode equipment.

When used in the Dexter 3000 platform, the user can access the powerful functionality of the **eXpedite**™ architecture: converged voice, video, and data wide-area network access; wire speed protocol interworking, IP routing and bridging; ATM quality of service for all data flows; and much more.

Key Benefits

- Enables customers to connect existing router and legacy equipment to a Frame Relay or ATM multiservice network without modification
- Allows connection of an ATM cell stream across clear channel circuits to be attached to ATM cell-switched networks without modification
- Provides a simple and cost-effective solution for Frame Relay across ATM transport services

Specifications

Facility Interface

• Connector:	DB25
• Bit rate:	Configurable between 64Kbps and 8Mbps
• Framing:	ATM cell or framed
• Line coding, Frame Relay Mode:	HDLC
• Clock source:	Internal or external
• Signal format:	RS-232, V.35, X.21

Frame Relay

- LMI T1.617 Annex D
- LMI Q.933 Annex A
- IP over Frame Relay per RFC 1490, Network and Service Interworking (FRF.5 and FRF.8)

Standards

- Electrical: EIA 530, V.35, X.21, RS-232
- Physical: EIA-530 DCE

MARINER NETWORKS
An Odetics Company

1585 South Manchester Avenue, Anaheim, CA 92802-2907
Tel 714-780-7685 Toll free 888-329-4487 Fax 714-780-7696
Sales 714-780-7987 E-mail sales@marinernetworks.com
Odetics Europe
Tel +44(0) 118 927 4607 E-mail ixb@odetics.co.uk Fax +44(0) 118 944 0069
www.marinernetworks.com www.dexteraccess.com

Chapter 2

Introduction

This section introduces the Dexter product and describes its principal functions and capabilities.

Dexter Overview

The Mariner Networks Dexter 3000 is a series of Integrated Access Devices (IADs), and forms part of the family of Dexter IADs and access concentrator communications products.

Dexter is a Customer Premises Equipment (CPE) solution that enables organizations to connect multiple branch offices economically to a multiservice ATM or Frame Relay Wide Area Network (WAN). It provides the means for branch end-users to combine their voice and data network connections on to a single low-speed network path, which can be more easily managed from the central headquarters.

Dexter connects to the customer's existing data, voice, and video equipment and resides in the end-user's communications room or closet. It is a sophisticated, branch-office, multiservice platform that provides many additional key functions and benefits over other CPE devices such as Frame Relay Access Devices (FRADs) or Time Division Multiplexers (TDMs).

Dexter can be configured as a host or CPE access device to provide:

- Frame Relay to ATM interworking
- Inverse Multiplexing over ATM (IMA) for up to 4 x E1/T1 lines
- Variable Bit Rate voice adaptation using AAL2 protocols
- Circuit Emulation Services using AAL1 protocols
- Comprehensive support for voice compression modulations

- Echo cancellation and silence suppression for AAL2 protocols
- Attachment to digital PBX using E1/T1 interfaces
- Analogue FXO and FXS operation with Ground or Loop Start
- E&M support for Types 1, 2, and 5 (Immediate, Delay, and Wink)
- Support for voice, video, or data over single or multiple ISDN-BRI
- IP routing and bridging over ATM
- DHCP and NAT support
- Comprehensive support for SNMP network management
- Maximum of 4096 connections (FR DLCIs, ATM VCs, etc.)
- ATM PCR, SCR, and MBS traffic shaping
- ATM classes: CBR, VBR-rt, VBR-nrt, UBR and UBR+
- ATM PVCs and SVCs
- Per port pacing
- Frame Relay QoS via DLCI : CIR
- Conformance to ATM and Frame Relay forum standards.

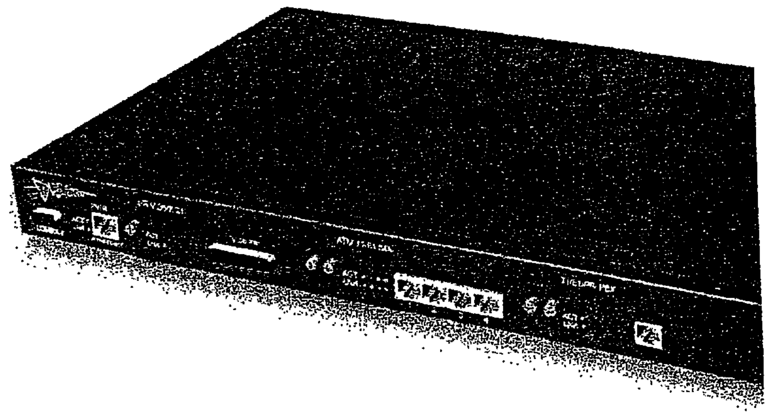


Figure 1 Dexter front panel view

Characteristics

Dexter has the following characteristics:

Interworking Technology

Dexter interworking solutions provide peer-to-peer connectivity between Dexters located in the branch offices and Dexters located in the central or regional office locations. ATM or Frame Relay PVCs or are mapped according to networking requirements, which provide for a fully meshed configuration to exist between all Dexters within a given Multiservice WAN.

Inverse Multiplexing over ATM

Dexter offers the capability of connecting up to 4 x 2Mbps circuits into a logical IMA group, thus allowing ATM PVCs or SVCs to utilize available bandwidth fully. In this mode, Dexter connects to the ATM WAN switch via multiple 1.5Mbps (T1) or 2Mbps (E1) leased lines. The adjacent ATM switch must be configured with an equal IMA facility to terminate the logical group prior to core network switching of cell traffic, or the IMA group can be carried intact across the WAN to another Dexter for termination.

Enhanced Voice Convergence

Dexter supports the multiplexing of compressed voice channels via ATM Adaptation Layer 2 (AAL2) protocols into a single ATM PVC or SVC, thus maximizing ATM bandwidth optimization. Further bandwidth efficiencies are obtained through utilizing silence suppression algorithms and local comfort noise generation to eliminate unnecessary cell transmissions. Additionally, Dexter supports uncompressed voice channel transmission via AAL1 structured Circuit Emulation Services (CES) to an adjacent Dexter or other vendor equipment.

IP Routing and Bridging

Dexter offers unparalleled performance versus cost using its proprietary technology to perform frame to cell conversion and data forwarding in hardware. Dexter performs both local IP routing (RIPv1 & v2) and switching as well as ATM bridging using multi-protocol encapsulation techniques over AAL5 (RFC 1483 and RFC 1577 for Classical IP). The bridging function also supports the Spanning Tree protocol.

Frame Relay to ATM Interworking

Local data connections are managed via Dexter's Frame Relay to ATM Interworking function. This facility enables customers to retain their existing router hardware and software configurations to preserve access to legacy applications. The data connection operates up to 2Mbps via a DB25 V.35X.21RS-530, or RS-449 interface. The interworking function supports either Network (FRF.5) or Service (FRF.8) Interworking in accordance with the Frame Relay Forum multi-protocol implementation agreements (RFC 1490

ATM Classes of Service

ATM PVCs and SVCsare fully supported to ATM UNI 3.0, 3.1, and 4.0 signaling. Quality of Service and traffic shaping per port is

provided via VCC PCR, SCR, and MBS parameters. Service classes are supported via Adaptation Layers 1, 2, and 5 utilizing classes CBR, VBR-rt, VBR-nrt, UBR, and UBR+.

Advanced Network Management

Dexter provides extensive network management facilities via its internal SNMP agent and a supporting SNMP Network Management Application. A full range of functions is available to configure, monitor, and report upon network performance, configuration parameters, call management, fault management, and IP/Frame Relay network protocol statistics.

Understanding Dexter Management

Dexter management is available through Mariner Networks' SNMPNetwork Management Application **Messenger**tm, which provides local and remote access to one or more Dexter IAD's via SNMP. The application is designed to provide the network management capabilities expected from enterprise or carrier-class customers. Network management is generally defined to encompass two main areas, namely Monitoring and Control.

- Network Monitoring is concerned with observing and analyzing the status and behavior of its network domain configuration and its devices.
- Network Control is concerned with the altering of parameters of various configurations of the network devices and causing those components to perform predefined actions.

In line with this concept, Dexter is a fully managed ATM IAD, which supports the following key disciplines:

- Network Management
- Traffic Management
- Code Management
- Security Management.

Network Management

You can manage Dexter's subsystems in any of the following ways:

- From an ASCII terminal with a character-based command line interface that is directly connected to the RS-232 console monitor port on Dexter's front panel.
- By remotely logging into the Dexter's Command Line Interface via a Telnet session. This session may be via the local Ethernet port, Frame Relay port, or in-band across the ATM WAN.
- By accessing Dexter's SNMP Agent via an authorized network management

station running Mariner Networks' SNMP Management Application "Messenger". The network management station may reside anywhere in the network.

Dexter's **Messenger**™ application can be run on any type of network management workstation irrespective of operating system or machine type. It can be run under HP OpenView™ or independently, offering a complete network management environment for the enterprise or carrier class user. The graphical user interface (GUI) enables the operator to configure Dexter elements quickly and easily and to interrogate performance data and traffic profiles in a variety of tables and charts. Multiple Dexter configurations and maps may be viewed simultaneously.

Dexter can support simultaneous access by multiple network management stations to facilitate redundancy and continuous network operational requirements. The SNMP agent comprises Dexter's enterprise MIB and a number of industry compliant networking MIBs (ATM, FR, and MIB-II).

Traffic Management

Dexter's advanced traffic management functions include:

- Priority queues per Quality of Service
- Constant Bit Rate (CBR)
- Real time Variable Bit Rate (VBR-rt)
- Non-real time Variable Bit Rate (VBR-nrt)
- Unspecified Bit Rate (UBR)
- Unspecified Bit Rate Plus (UBR+)
- Traffic shaping per port and per Virtual Circuit (TM 4.0).

Dexter ensures that the Virtual Channel Connection (VCC) contract is respected at the Virtual Channel (VC) level. To reduce irregular bursts of traffic, a reshaping function is provided.

Code Management

Code management allows the network administrator or network operator to manage the application and user configuration modules contained within Dexter. The application module contains the program logic necessary for Dexter to function. User configuration modules consist of parameters and network definitions that describe the network, voice characteristics, profiles, and packet/cell routing information.

Dexter's flash memory can hold multiple copies of application modules as well as multiple copies of user configurations, and allows an operator to switch between them. In this way, Dexter can be reloaded or re-configured to perform differently while still retaining the ability to recover from updates that fail to function as

required.

You can access Dexter's code management in any of the following ways:

- Application and user configuration module data can be uploaded or downloaded using TFTP. Dexter contains a TFTP server that enables bi-directional processes.
- Switching between application or user configuration data can be performed using either the console port via the command line interface (CLI), via a Telnet session, or remotely via the Management application.
- Using the console monitor port, you can perform uploading and downloading of application or user configuration data.

Providing multiple copies of application and user configuration data in flash memory enhances Dexter's network manageability in a customer premises environment. Dexter's advanced network management capabilities enable network control and monitoring to be performed quickly and simply with the minimum of end-user involvement.

Security Management

Dexter is configured with the following security features:

Configuration Protection

Access to Dexter via the console monitor port is password protected. This password can be changed at the customer's/end-user's discretion. A hardware-based reset feature is incorporated to enable recovery to a default password in the event of password loss.

Network Access Protection

Telnet access to Dexter's Command Line Interface (CLI) via the ATM, local Ethernet or Frame Relay network is provided and access is controlled via a password.

Access to the Dexter SNMP agent is controlled via a domain name to prevent and limit unauthorized use.

Typical Implementations

Dexter simplifies ATM access at the customer premises. This is achieved through implementing Dexter as an ATM Interworking Network Terminating Unit (NTU) that clearly defines the boundary of the ATM network from the customer's local network communications equipment. Through its ATM interworking capabilities, Dexter converges multiple services (voice, data, and

video) over single or multiple upstream ATM links. The following diagram illustrates a typical configuration.

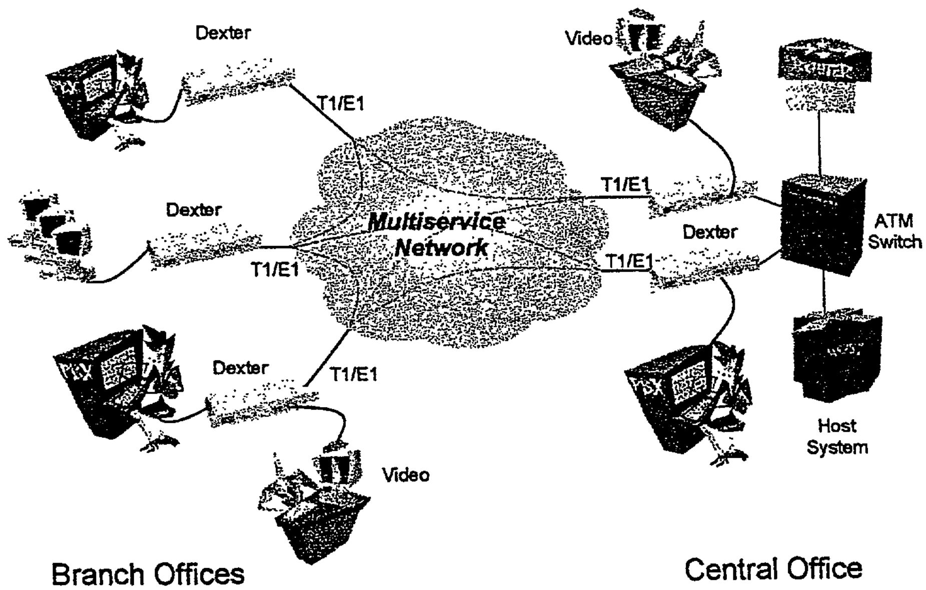


Figure 2 Typical Dexter configuration

Figure 2 illustrates a simple "mesh system" implemented between several office locations. All Dexters are configured to establish PVCs between remote locations and to the central location housing the host system and application servers. Multiple Dexters may be installed at the central location to provide sufficient voice channel capacity for head office personnel.

This page left intentionally blank

Chapter 3

Features and Components

This chapter describes the main features and components of Dexter.

Base Product

The base Dexter 3000 product consists of a 3-slot chassis enclosure with the following components:

- Main processor board with application software loaded
- Power supply assembly
- 1 x RJ45 Ethernet port
- 1 x DB9 RS-232 console monitor port
- Three blank single-slot filler plates.

The following components are delivered with Dexter to facilitate power up and initial configuration:

- Power supply cord
- RS232 modem cable
- Documentation CD-ROM package.

System Component

All Dexter units are based upon a main processor board design and chassis enclosure that facilitates the insertion of up to three Network or User Interface Modules. Available modules are described later in this chapter.

The main processor board contains the CPU, various memory modules, operating system, and application code. Additionally, this board holds Mariner Networks' **eXpedite**^{tm} processor, a proprietary technology that performs frame to cell conversion and data forwarding in hardware. This feature is described more fully in Chapter 4.

Ethernet Port

Dexter is equipped with a single RJ45 socket on the front panel system unit to facilitate either 10BaseT Ethernet or Telnet

management access. In this way, Dexter can be configured without the need for any modules to be inserted prior to customer delivery. Initial configuration of IP addressing would need to be achieved via the console monitor port.

Console Monitor Port

Dexter is equipped with the DB9 RS-232 female DCE connector on the front panel system unit to facilitate initial configuration of the Dexter unit.

Memory Configuration

16MB of main memory is provided in all Dexter configurations. In addition to this memory offering, Dexter is configured with flash memory to hold multiple application and user configuration data, and boot PROM to support initial power-on and program load functions.

Power Requirement

Each unit is configured with an internal auto-detecting 85-264 VAC power supply with a fused power switch. A power cord applicable to the destination country is included.

Documentation

A printed Quick Start Installation Guide is delivered with all Dexter units. All other documentation relating to Dexter is available on an accompanying CD-ROM. Additionally, all user-related documentation is available by downloading from the Mariner Networks website.

Cabling

Each Dexter is shipped with an RS-232 DB9 DCE/DTE modem cable. Access to the console monitoring port is through a VT100 terminal device.

Additional Required Components

In addition to the base components supplied with the chassis, Dexter will need to be populated with one or more Network or User Interface Modules that connect the ATM WAN or the existing customer communications equipment. A brief description of each module is contained within this document. Further detailed information can be obtained by referencing the relevant Dexter 3000 Supplementary Data Sheet.

Optional Components

The following optional components are available.

Mounting Kits

Dexter is designed to be either a standalone, wall-mounted, or rack-mounted unit. Mounting kits are available to facilitate the installation of the Dexter unit into a 19 inch communications rack or onto a wall.

Cabling

A number of cabling options is supported to accommodate connection of the ATM interface, and Frame Relay V.35/X.21 attached router. Further cabling specifications are available upon request.

Documentation

All documentation relating to Dexter is available on a CD-ROM. This CD-ROM is shipped with Dexter or can be ordered separately. Please refer to the Mariner Networks website for further information on available publications.

Modules

This section describes the various network and user modules available for the Dexter IAD. The following list provides a summary functional description.

Module	Description	Page
T1/E1	Network module to connect to ATM or Frame Relay WAN.	3-5
ATM T1 or E1 IMA	Network module to connect to ATM WAN. Supports grouping of multiple ATM links into single VC.	3-6
ATM DS-3/E3	Network module to connect to ATM WAN.	3-7

ATM OC-3 or STM-1	Network module to connect to ATM WAN.	3-8
SDSL	Network module to connect to ATM or Frame Relay WAN over DSL.	3-9
HDSL2	Network module to connect to DSL WAN. Configurable for ATM or Frame Relay.	3-10
Synchronous Serial	Network or User module to connect router or other Frame Relay device.	3-11
10/100BaseT	User module used to connect local Ethernet hub or switch.	3-12
Voice T1/E1/PRI	User module to connect local PBX equipment.	3-13
Voice T1/E1/PRI + BRI	User module to connect local PBX and ISDN BRI	3-15
ISDN BRI	User module to connect up to 3 S/T/U devices.	3-16

Table 1 Module list description

T1/E1

There are two types of T1/E1 network module available for Dexter:

- 1 x port T1/E1
- 4 x port T1/E1 announcement to be defined (TBD).

Each module may be configured to operate in ATM cell delineation or Frame Relay HDLC delineation mode. Each interface is presented as an RJ48C female socket that can accept either a T1 (1.5Mbps) or an E1 (2Mbps) facility interface.

Each module has the following characteristics:

- 1 or 4 ports each operating at either 1.544Mbps or 2.048Mbps line rate.
- Each port may connect to an ATM switch via UNI (3.0, 3.1, or 4.0), or a Frame Relay DLCI compliant device.
- Integrated CSU/DSU functionality.
- Physical interface is electrical with impedance of 100/120 Ohms.
- One or more modules may be inserted into any of Dexter's slots.
- Both modules are easily swappable without the need for specialist knowledge or equipment. Dexter will require rebooting and reconfiguring upon change of module type.

Figure 3 shows both module faceplates.

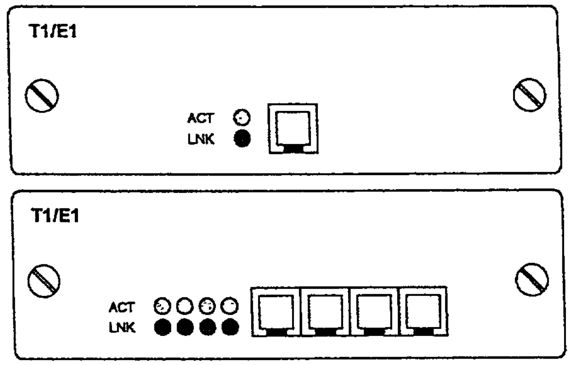


Figure 3 T1/E1 modules

ATM T1/E1 IMA

There is one type of the ATM Inverse Multiplexing over ATM (IMA) network module available for Dexter:

- 4 x port E1 or T1.

This module may be configured to operate in a variety of logical IMA line groups. Each interface is presented as an RJ48C female socket that can accept either a T1 (1.5Mbps) or E1 (2Mbps) facility interface.

The module has the following characteristics:

- 4 ports, each operating at either 1.544Mbps or 2.048Mbps line rate.
- Each port may connect to an ATM switch via UNI using a supported interface.
- T1 option has an integrated CSU/DSU functionality.
- Physical interface is electrical with impedance of 100/120 Ohms.
- One or more modules may be inserted into any of Dexter's slots.
- This module is easily swappable without the need for specialist knowledge or

equipment. Dexter will require rebooting and reconfiguring upon change of module type.

Figure 4 shows the faceplate of the module.

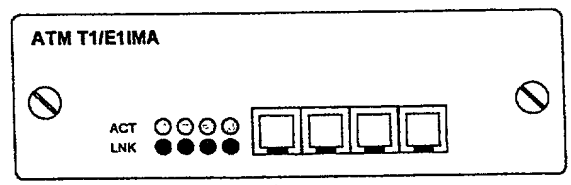


Figure 4 ATM T1 or E1 IMA module

ATM DS-3/E3

There are two types of a ATM DS-3/E3 network module available for Dexter:

- 1 x port DS-3 announcement TBD
- 1 x port E3 announcement TBD.

Each module is configured to operate in ATM cell delineation mode. Each interface is presented as a BNC 75 Ohm female connector that can accept either a DS-3 (45Mbps) or an E3 (34Mbps) facility interface.

Each modules has the following characteristics:

- 1 port operating at 34Mbps or 45Mbps line rate.
- Each port may connect to an ATM switch via UNI using a supported interface.

- Physical interface is electrical with an impedance of 75 Ohms.
- One or more modules may be inserted into any of Dexter's slots.
- Both modules are easily swappable without the need for specialist knowledge or equipment. Dexter will require rebooting and reconfiguring upon change of module type.

Figure 5 shows both module faceplates.

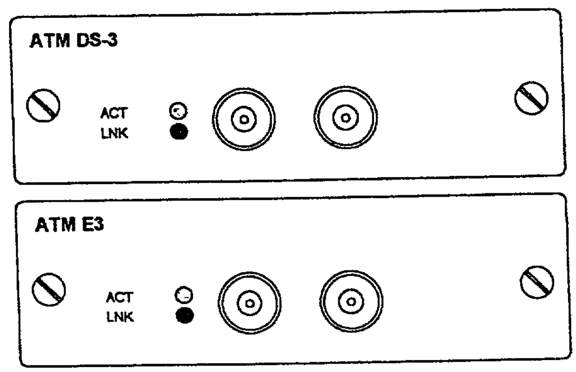


Figure 5 ATM DS-3/E3 modules

ATM OC-3/STM-1

There are two types of ATM OC-3/STM-1 network module available for Dexter:

- 1 x port OC-3 announcement TBD
- 1 x port STM-1 announcement TBD

Each module is configured to operate in ATM cell delineation. The interface is presented as an optical fiber ST female connector that can accept either an OC-3 (155Mbps) or STM-1 (155Mbps) facility interface.

The module has the following characteristics:

- 1 port operating at 155Mbps line rate software configurable between either

the OC-3 or STM-1 format.

- Each port may connect to an ATM switch via UNI using a supported interface.
- Physical interface is single or multimode optical fiber.
- One or more modules may be inserted into any of Dexter's slots.
- This module is easily swappable without the need for specialist knowledge or equipment. Dexter will require rebooting and reconfiguring upon change of module type.

Figure 6 shows the module faceplate.

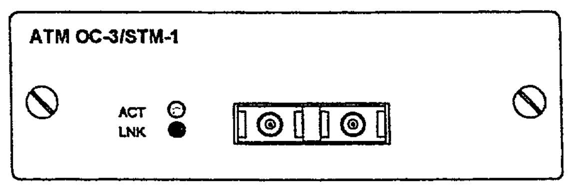


Figure 6 ATM OC-3/STM-1 module

SDSL

There is one type of the SDSL network module available for Dexter:

- 2 x port SDSL announcement TBD

The module may be configured to operate in ATM cell delineation or Frame Relay delineation mode. The module may be configured to communicate with another Dexter, DSLAM or other Central Office (CO) equipment. The module can be configured as either a CO or CPE device.

The module has the following characteristics:

- 2 ports operating in variable rate SDSL (Symmetric Digital Subscriber Line) using Globespan's™ 2B1Q XDSL chip set. SDSL data rates of 144kb/s, 272kb/s, 400kb/s, 528kb/s, 784kb/s, 1040kb/s, 1168kb/s, 1552kb/s, 2064kb/s, and 2320kb/s are supported using 2B1Q line encoding data rates.
- Each port may connect to an ATM switch via UNI, or a Frame Relay compliant device.
- Physical interface is electrical with impedance of 50/75 Ohms. The connectors are RJ11 terminating voice grade telephone wire local loops.
- One or more modules may be inserted into any of Dexter's slots.
- This module is easily swappable without the need for specialist knowledge or equipment. Dexter will require rebooting and reconfiguring upon change of module type.

Figure 7 shows the module faceplate.

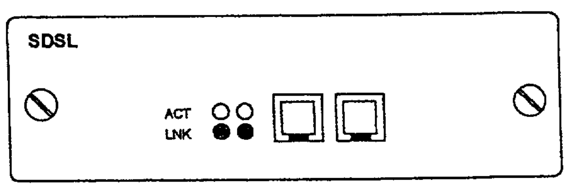


Figure 7 SDSL module

HDSL2

There is one type of HDSL2 network module available for Dexter:

- 1 x port ATM/FR announcement TBD

The module may be configured to operate in ATM cell delineation

or Frame Relay delineation mode. The module may be configured to communicate with another Dexter, DSLAM, or other Central Office (CO) equipment. The module can be configured as either a CO or CPE device.

The module has the following characteristics:

- 1 port operating up to 1.5Mbps using 2B1Q line encoding data rates.
- The port may connect to an ATM switch via UNI, or a Frame Relay compliant device.
- Physical interface is electrical with impedance of 50/75 Ohms. The connector is RJ11 terminating voice grade telephone wire local loops.
- One or more modules may be inserted into any of Dexter's slots.
- This module is easily swappable without the need for specialist knowledge or equipment. Dexter will require rebooting and reconfiguring upon change of module type.

Figure 8 shows the module faceplate.

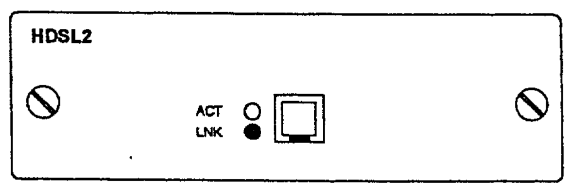


Figure 8 HDSL2 module

SYNCHRONOUS SERIAL

There is one type of the Synchronous Serial module available for

Dexter:

- 1 x port

The module is configured to operate in Frame Relay mode, clear channel or channelized mode, or ATM mode via clear channel. The module can attach to an existing Frame Relay router or other Frame Relay compliant device. The interface can be configured for either V.35 or X.21 via an adapter cable..

The module has the following characteristics:

- 1x DB25 female DCE/DTE synchronous port supporting, RS-530, or RS-449. Data rate can be set from 64K to 8.192Mbps, full duplex operation.
- One or more modules may be inserted into any of Dexter's slots.
- The module is easily swappable without the need for specialist knowledge or equipment. Dexter will require rebooting and reconfiguring upon change of module type.

Figure 9 shows the module faceplate.

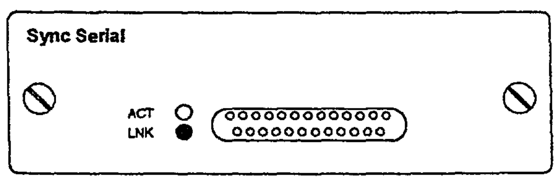


Figure 9 Synchronous Serial module

10/100BaseT

There is one type of the 10/100BaseT module available for Dexter:

- 4 x port 10/100BaseT announcement TBD

The module is configured to attach to an existing Ethernet LAN via a hub or switch. Each RJ45 port is rate auto-sensing and provides either switching of Ethernet packets between Dexter's LAN interfaces or routing/bridging via AAL5 encapsulation over the ATM WAN.

The module has the following characteristics:

- 4 ports of 10/100BaseT for local Ethernet or Telnet management access.
- Spanning Tree protocol is supported.
- Each port is on its own segment.
- One or more modules may be inserted into any of Dexter's slots.
- The module is easily swappable without the need for specialist knowledge or equipment. Dexter will require rebooting and reconfiguring upon change of module type.

Figure 10 shows the module faceplate.

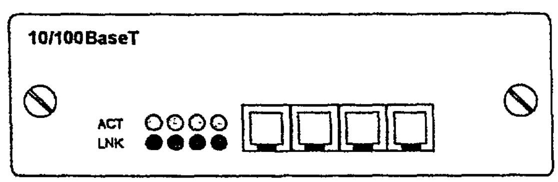


Figure 10 10/100BaseT module

VOICE T1/E1/PRI

There is one type of Voice T1/E1/PRI module available for Dexter:

- 1 x port T1/E1

The module may be configured to operate in either T1 or E1 mode and connects to the customer's local PBX system. The module provides a T1/E1 trunk type interface that can support either 24 (T1) or 30 (E1) channels of voice throughput. PBX supported interface signaling includes either Robbed Bit (T1), CAS (E1), or ISDN PRI using Common Channel Signaling (CCS) to provide 23 (T1) and 30 (E1) bearer channels respectively for voice trunking. The module also contains the necessary Digital Signal Processors (DSPs) and logic to provide voice compression, silence suppression, echo cancellation, AAL1AAL2 processing, and packet to cell conversions.

The module has the following characteristics:

- 1 port operating at either 1.544Mbps (T1) or 2.048Mbps (E1). The module can be ordered with support for 8, 16, 24, or 32 voice channels. These channels may be assigned to any time slot in the T1 or E1.
- Signaling supported includes RBS, CAS (E1) and ISDN PRI (CCS).
- Supported CCS signaling for ISDN PRI includes PRI Net5 User, PRI Net5 Network, and PRI QSIG.
- AAL1 voice processing in accordance with af-vtoa-0078.000.
- AAL2 voice processing in accordance with ITU-T I.363.2.
- Voice processing includes G.711 (64K PCM), G.726 ADPCM, G.727 EADPCM, G.729 CS-ACELP, G.729AB CS-ACELP, and G.723.1A.
- Support for Fax Relay and voice-band signaling.
- Physical interface is an RJ45 electrical with impedance of 100/120 Ohms.
- One or more modules may be inserted into any of Dexter's slots.
- The module is easily swappable without the need for specialist knowledge or equipment. Dexter will require rebooting and reconfiguring upon change of module type.

Figure 11 shows the module faceplate.

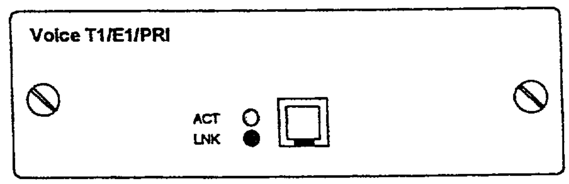


Figure 11 Voice T1/E1/PRI module

VOICE T1/E1/PRI + BRI

There is one type of Voice T1/E1/PRI + BRI module available for Dexter:

- 1 x port T1/E1 + 1 x port ISDN BRI

The PBX T1/E1 facility interface operates identically as outlined in the previous module. Additionally, this module incorporates an ISDN BRIport that provides for attachment to a videoconferencing codec (although it may be used with any ISDN BRI compliant device).

The module has the following characteristics:

- 1 port operating at either 1.544Mbps (T1) or 2.048Mbps (E1). The module can be ordered with support for 8, 16, 24, or 32 voice channels.
- Identical characteristics to that of the PBX E1/T1 module.
- 1 ISDN BRI port providing 2 x 64K bearer channels and 1 x 16K D channel. Both S/T and U interfaces are supported.
- One or more modules may be inserted into any of Dexter's slots.
- The module is easily swappable without the need for specialist knowledge or equipment. Dexter will require rebooting and reconfiguring upon change of module

type.

Figure 12 shows the module faceplate.

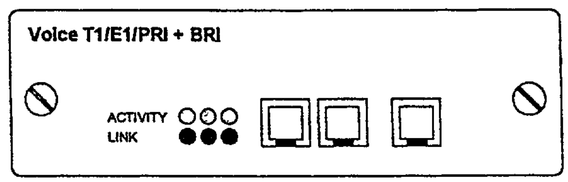


Figure 12 Voice T1/E1/PRI + BRI module

ISDN BRI

There are two versions of ISDN BRI module available for Dexter:

- 2 x port ISDN BRI announcement TBD
- 3 x port ISDN BRI announcement TBD

This module is equipped with either a dual port or triple port ISDN BRI facility that supports S/T and U interfaces. Each port can be configured to support voice, fax, or voice-band data signals. Full voice processing is supported for compressed or uncompressed transmission across the ATM WAN.

Each version of the module has the following characteristics:

- 2 or 3 ports providing ISDN BRI service. Each port supports 2 x 64K bearer channels and 1 x 16K D channel. Both S/T and U interfaces are supported.
- One or more modules may be inserted into any of Dexter's slots.
- Both modules are easily swappable without the need for specialist knowledge or equipment. Dexter will require rebooting and reconfiguring upon change of module

type.

Figure 13 shows both module faceplates.

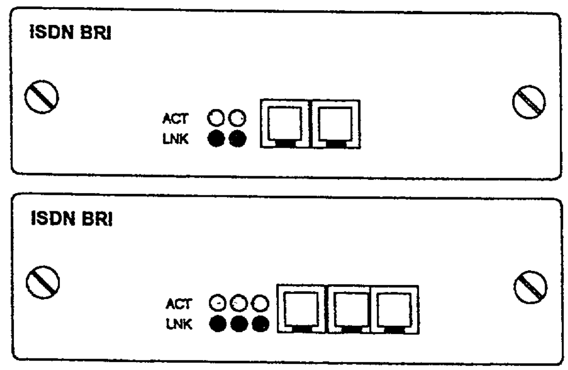


Figure 13 ISDN BRI modules

Chapter 4

Functional Description

This chapter describes the functions of each main component in Dexter and provides an overview of various communication technologies.

Architecture Overview

Dexter comprises of a main processing board that contains core memory, application code, and optional interface modules. A key element of this design is Mariner Networks' proprietary processor technology, **eXpedite**™.

The **eXpedite**™ processor consists of a cell switching fabric with segmentation and re-assembly processes and a cell forwarding architecture that includes a cell scheduler function. It contains the necessary logic and dynamic tables to translate between ATM VCs and Frame Relay DLCIs. Additionally, through its powerful scheduling ability, it supports current ATM and Frame Relay Quality of Service (QoS) attributes. The **eXpedite**™ processor uses Dexter's on-board CPU to build and maintain its tables and routing information.

The **eXpedite**™ processor's unique benefit is that once its tables have been defined, it converts, routes, and switches frames and cells effortlessly, in hardware, and releases the main CPU to perform other processor intensive tasks such as voice processing. Unlike other comparable CPE devices, this blend of technology enables Dexter to deliver the processing power and switching performance that would normally be found in larger and more expensive access units.

Dexter's other key components are the following subsystems:

- ATM Processing
- Voice Processing
- Network Management.

ATM Processing

This subsystem provides the broadband services to Dexter's applications.

Overview

ATM processing, frame to cell conversion and transmission of cells to the ATM network modules is performed by the **eXpedite**™ processor.

The following ATM Adaptation Layers (AAL) and associated service classes are supported:

Layer	Service Class	Mnemonic

AAL1	Constant Bit Rate	CBR
AAL2	Variable Bit Rate	VBR-rt VBR-nrt
AAL5	Unspecified Bit rate	UBR UBR+

Table 2 Supported AAL protocols

AAL1 Operation

This layer is used to support all switched or permanent uncompressed voice calls. Uncompressed voice traffic is either carried as a structured or basic Nx64K CES cell stream as defined in the af-vtoa-0078.000 interoperability specification, Circuit Emulation Services (v2).

AAL2 Operation

This layer is used to support all switched compressed voice calls over the ATM network. All AAL2 voice traffic between a pair of Dexters is multiplexed across a single ATM VC.

AAL5 Operation

This layer is used to support all Frame Relay data frames and Internet data packets over the ATM network.

Quality of Service

Dexter performs traffic shaping of its outgoing ATM cell flow in accordance with the relevant standard for Connection Traffic Descriptor that was negotiated with the ATM network. The relevant parameters used to specify unambiguously the conforming cells of the ATM connection are Peak Cell Rate (PCR), Sustainable Cell Rate (SCR), and Maximum Burst Size (MBS). Dexter contains two leaky buckets to support its QoS scheduling.

Inverse Multiplexing over ATM Interface

Dexter can be configured to accept up to 4 x 2Mbps circuits via a 4-port E1/T1 IMA interface, which can be configured into two IMA logical groups. Typically, ATM PVCs would utilize all available circuits in the IMA group to provide greater throughput. An outline flow of ATM cells through an IMA configuration is illustrated in Figure 14. Here, an ATM data stream is split across three individual physical links on a cell-by-cell basis in a "round-robin" effect.

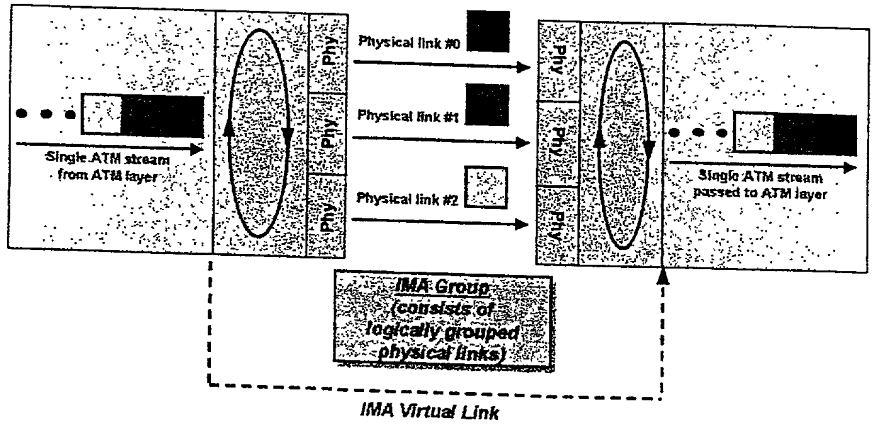


Figure 14 IMA logic flow

Frame Relay to ATM Operation

Dexter supports both Frame Relay to ATM "Network" and "Service" interworking as defined by the Frame Relay Forum's Frame Relay/ATM Network and Service Interworking Implementation Agreements (FRF.5 and FRF.8 respectively).

Network Interworking (FRF.5)

This function is responsible for forwarding frames between the Frame Relay interface and the ATM Data Subsystem. Dexter processes frames received from the Frame Relay interface as follows:

1. De-multiplexed according to their DLCI.
2. Stripped of their HDLC encapsulation headers.
3. BECN, FECN, and DE congestion and flow control indicators are mapped according to ATM EFCI and CLP settings.
4. Re-encapsulated in ATM AAL5 CPCS PDUs.
5. Segmented and multiplexed over the UTOPIA cell interface according to the ATM VCC.

In the reverse direction, the ATM cell traffic is processed as follows:

1. ATM AAL5 CPCS PDUs reassembled from the UTOPIA cell interface.

2. De-multiplexed according to the ATM VCC.
3. Stripped of their AAL5 encapsulation overhead bytes.
4. ATM EFCI, DE congestion, and flow control indicators are mapped according to FR BECN, FECN, and DE settings.
5. Multiplexed over the appropriate Frame Relay interface according to DLCI.

Figure 15 illustrates Network interworking mapping performed between frames and cells.

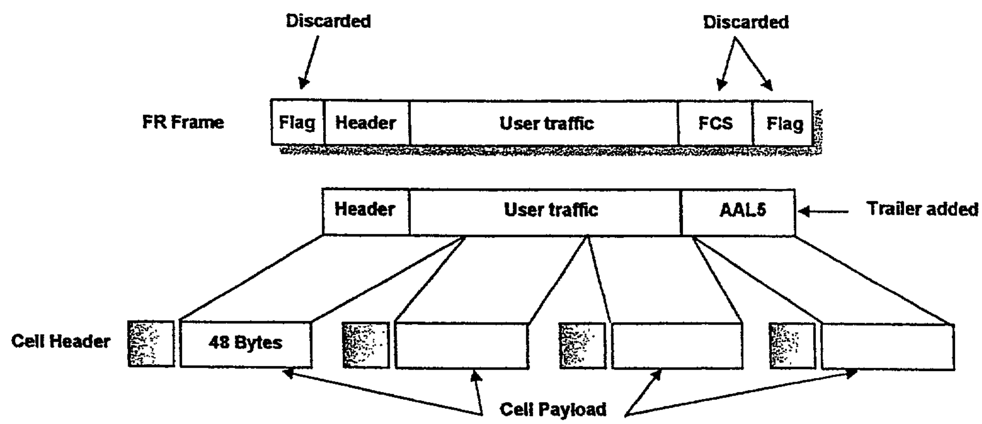


Figure 15 Network interworking mapping

Service Interworking (FRF.8)

This function is essentially the same as the previous function, except that protocol conversion algorithms are applied to convert Frame Relay bridged or routed PDU to ATM bridged or routed PDUs. Frames received from the Frame Relay interface are processed as follows:

1. De-multiplexed according to their DLCI.
2. Stripped of their HDLC encapsulation headers.
3. Network protocol encapsulation headers mapped from those specified in RFC 1490 (for Frame Relay) to those

specified in RFC 1483 (for ATM).

4. Re-encapsulated in ATM AAL5 CPCS PDUs.
5. Segmented and multiplexed over the UTOPIA cell interface according to the ATM VCC.

In the reverse direction, Dexter processes the ATM cell traffic as follows:

1. ATM AAL5 CPCS PDUs reassembled from the UTOPIA cell interface.
2. De-multiplexed according to the ATM VCC.
3. Stripped of their AAL5 encapsulation overhead bytes.
4. Network protocol encapsulation headers mapped from those specified in RFC 1483 (for ATM) to those specified in RFC 1490 (for Frame Relay).
5. Multiplexed over the appropriate Frame Relay interface according to DLCI.

Figure 16 illustrates Service Interworking mapping performed between frames and cells.

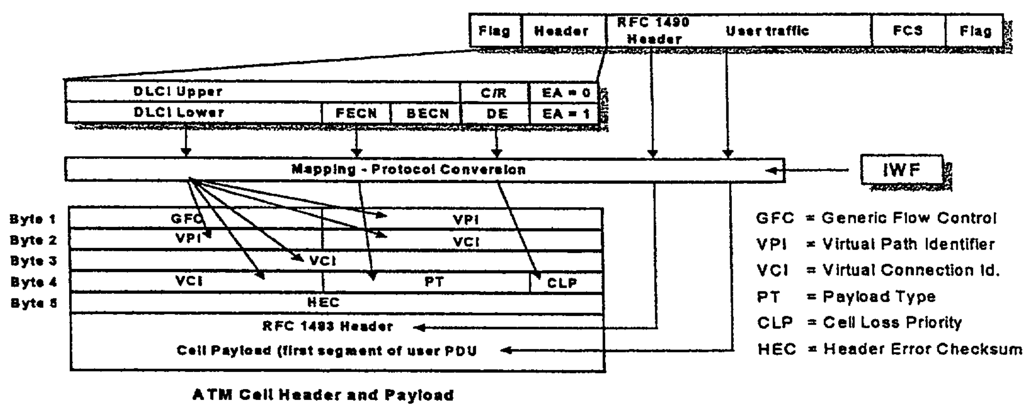


Figure 16 Service interworking mapping

Ethernet Operation

Dexter is assigned an IP address and subnet mask to each network port (including ATM WAN ports). Services such as Domain Host Control Protocol (DHCP) and Network Address Translation (NAT) are supported.

Dexter performs both local IP routing (RIPv1 & v2) and switching between its local and network ports. Bridging between a pair of

Dexters is achieved by using ATM bridging multi-protocol encapsulation techniques over AAL5 (RFC 1483) and Classical IP encapsulation (RFC1577).

Other protocols built into the Dexter IP stack include UDP, TCP, TFTP, SNMP, ARP, and ICMP. Telnet packets received from the local ports or via the network ports are converted to command strings and passed to the Dexter's command line interface (CLI) for parsing.

Domain Host Configuration Protocol

The Dynamic Host Configuration Protocol's (DHCP) purpose is to enable individual computers on an IP network to extract their configurations from a server (the 'DHCP server') or servers, and in particular, servers that have no exact information about the individual computers until they request the information. The overall purpose of this is to reduce the work necessary to administer a large IP network. Dexter contains a DHCP server function

Network Address Translation

Network Address Translation (NAT) is used to translate one IP address to another. NAT can be used to allow multiple PCs to share a single Internet connection. It can also be used as a security tool by shielding the IP addresses of devices within the attached intranet. NAT can also be used for general IP address management by protecting the attached intranet from excessive address changes due to other network addressing constraints.

Voice Processing

This subsystem provides the voice and video-oriented narrowband services to Dexter's applications.

Overview

This section describes the functional aspects of Dexter's voice processing capabilities. Dexter's voice traffic across the ATM WAN is managed using a mixture of both AAL1 CBR connections and AAL2 VBR-rt connections.

AAL1 is used to carry uncompressed voice channels and associated Robbed Bit or CAS signaling transparently, end-to-end. AAL2 is used in conjunction with Mariner Networks' proprietary signaling and compression engine to switch and carry packetized, compressed voice traffic end-to-end. The AAL type is software configurable on a trunk channel basis, and compression algorithm/ratio basis.

Circuit Emulation Services

Dexter utilizes structured Circuit Emulation Services (CES), nailed up circuits supporting Nx64K (uncompressed) between Dexters, or between Dexter and other vendors' equipment supporting standards-based CES. While uncompressed CES-based connections are less efficient than compressed, AAL2 based connections, they offer the greatest benefit in terms of end-to-end voice quality and interoperability.

Figure 17 illustrates some of the network interconnection scenarios that can be implemented using structured circuit emulation with a Dexter network.

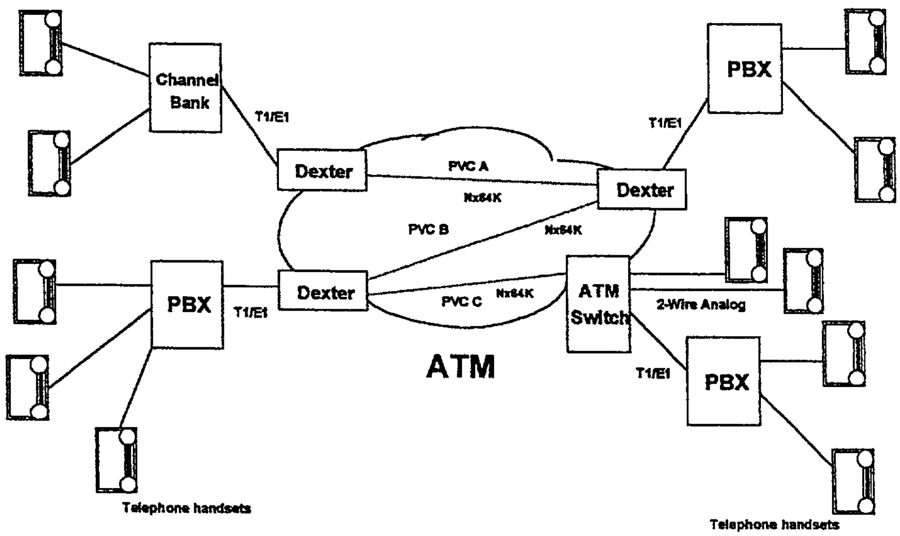


Figure 17 Dexter CES-based voice connections

In Figure 17, each of the ATM PVCs shown (A, B, C) carries a fixed, constant bit rate stream of ATM cells. The cell payloads, formatted according to the rules specified in af-vtoa-0078.000, contain voice samples and robbed bit signaling information for the trunk channels that the associated PVCs are configured to transport between the attached voice interfaces and the ATM network.

A CES connection provides a "nailed-up" transport for TDM voice data and voice signaling, allowing geographically dispersed telephony endpoints to communicate transparently over the ATM

network.

Circuits can be configured for either "Basic Mode", meaning that trunk channels are transported without associated signaling, or CAS mode, meaning that CAS/robbed bit signaling information is included in the cell payloads. The latter is useful for connecting non-PBX type equipment (e.g., analog handsets) at one end to PBX/trunk terminating equipment at the other end (loop extension).

Compressed Voice Services

By using AAL2 VBR-rt ATM circuits in conjunction with Dexter's compression and signaling software, Dexter can more efficiently transport voice and fax traffic across the ATM WAN.

AAL2 provides for the bandwidth-efficient transmission of low-rate, short, and variable packets in delay sensitive applications. ATM's VBR-rt services enable statistical multiplexing for the higher layer requirements demanded by voice applications, such as compression, silence detection/suppression, and idle channel removal. Additionally, in contrast to AAL1 (which has a fixed payload), AAL2 offers a variable payload within cells and across cells.

Mariner Networks' compression and signaling software terminates the local signaling channels and provides inter-Dexter proxy signaling over AAL5. This signaling provides for compressed calls that includes Robbed Bit/CAS modes, and out-of-band Common Channel Signaling (CCS) for a number of message oriented signaling protocols.

Initial releases of Dexter support compressed calls with in-band signaling (Robbed Bit/CAS) for non-ISDN T1/E1 interfaces and the following CCS variants when Dexter is configured for ISDN PRI mode:

- PRI Net5 User Mode
- PRI Net5 Network Mode
- PRI QSIG.

Figure 18 illustrates some of the network interconnection scenarios that can be implemented using a network of Dexters and voice compression and multiplexing technologies.

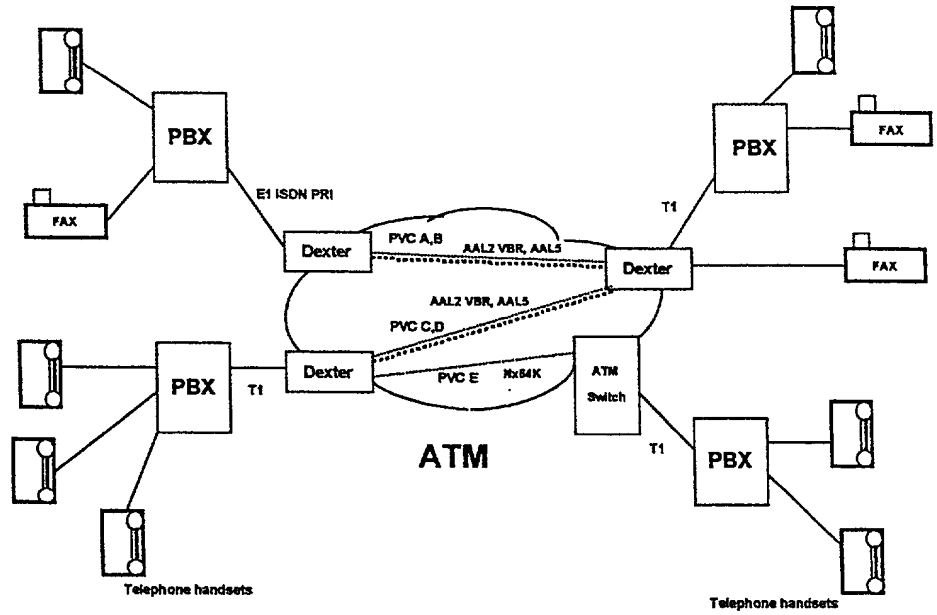


Figure 18 Dexter CES and AAL2-based voice connections

Figure 18 has the following key attributes:

1. Any combination of AAL1 uncompressed and AAL2 compressed calls can be configured and carried by Dexter.
2. In addition to an AAL2 VCC between a pair of Dexters, an AAL5 signaling VCC is required to carry Dexter's proprietary signaling protocol for switched, compressed voice/fax calls.
3. Inter-Dexter AAL2 compressed VCCs can be used to connect dissimilar PBX technologies (e.g., ISDN PRI using CCS to standard T1 using robbed bit signaling).
4. Dexter can also support analog interfaces that directly interface to fax machines, emulating the functions of a PBX to the attached devices.

Protocols and Standards Compliance

Dexter implements a combination of both standards-based and non-standards-based software protocols. The following sections provide an overview of these protocols.

AAL1

Dexter implements Nx64K structured mode CES over AAL1, as defined in af-vtoa-0078.000. Dexter is software configurable, on a

per-VCC basis, to run either Basic or CAS-mode CES for configured trunk channels. Trunk channels carried via CES are transported in uncompressed, 64K PCM format. Dexter does not implement unstructured mode CES (as defined in af-vtoa-0078.000), nor does it implement SRTS clock recovery as defined for AAL1 transport by the ATM Forum and ITU.

AAL2

Dexter implements a software based AAL2 implementation that is proprietary. This implementation utilizes the "general framework and Common Part Sublayer (CPS)" of the AAL type 2 defined in ITU-T Recommendation I.363.2. The associated cell payloads comprise compressed voice/fax data output by the Dexter compression engine.

It is Mariner Networks policy to implement standards-based software solutions wherever possible to maximize interoperability opportunities. Once the standards for AAL2 signaling have been agreed and accepted, Mariner Networks will implement such solutions into Dexter's AAL2 voice processing software.

AAL5

Dexter implements the ITU-T I.363.5-compliant AAL5 UBR transport mechanisms widely deployed today. This service is used to convey Dexter voice signaling messages in conjunction with AAL2-based voice traffic.

Voice Compression

Voice compression is performed by Dexter's compression engine that consists of software logic and a number of Digital Signaling Processors (DSPs). Dexter can be configured to operate with a maximum of 4 DSPs. Each DSP can support the processing of 8 voice channels concurrently. Dexter can be configured to support any set of the following voice encoding techniques:

- G.711 PCM, 64Kbps
- G.726 ADPCM, rates 16, 24, 32, and 40Kbps
- G.727 EADPCM, rates 16, 24, 32, and 40Kbps
- G.729A CS-ACELP and G.729B CS-ACELP, 8kbps rate
- G.723.1A, rates 5.3 and 6.3Kbps.

Proprietary Protocols

As the ATM Forum and/or the ITU do not yet standardize signaling for AAL2, Mariner Networks utilizes Dexter's proprietary signaling protocol to establish and tear down individual compressed voice calls. These calls are signaled using Robbed Bit/CAS/CCS modes

on the facility side, and converted to/from Dexter's proprietary "Q.931-like" signaling protocol for managing inter-Dexter call states.

PBX Interface Mode

Dexter can operate in one of three modes: North American T1, Standard E1, and E1-based ETSI ISDN PRI.

In T1 mode, narrowband signaling is via the AB bit transitions in robbed bit frames of the T1 Super Frame (SF) or Extended Super Frame (ESF) multiframe. In E1 (non PRI) mode, narrowband signaling is via CAS AB bit transitions in slot 16 of all frames in the E1 (FAS/CAS or FAS/CAS-CRC4) multiframe. In E1 PRI mode, narrowband signaling is configurable as QSIG, PRI NET5 User Side, or PRI NET5 Switch Side, via CCS in timeslot 16 of all frames in the E1 (FAS/CAS or FAS/CAS-CRC4) multiframe.

Trunk Channel Signaling

Dexter supports the following narrowband signaling protocols for trunk channel signaling. For each channel, one of the following may be selected as the signaling protocol:

- Foreign Exchange Station Loop Start or Ground Start
- Foreign Exchange Office Loop Start or Ground Start
- E&M Immediate Start
- E&M Delay Start
- E&M Wink Start.

This operation is unavailable when Dexter is operating in PRI mode.

Voice Coding Profiles

PCM voice samples from the PBX interface are switched through Dexter's on-board Digital Signaling Processors (DSPs), on a per-call basis, in order to perform the required compression, silence suppression, voice activity detection, and echo cancellation processes. All DSPs (up to a maximum of 4) are loaded with the same image at power up, which supports the following protocols (on a per channel basis, 8 channels per DSP):

- G.711
- G.729A and B
- G.726
- G.727
- Standard Fax relay.

Configuration of the DSP feature set is achieved through the creation of "Voice Coding Profiles". A coding profile is a set of configuration parameters that is assigned to a compressed call. The information in the coding profile informs the DSP how to process and route the compressed call through the system.

Coding profiles with common characteristics must be configured on both Dexter peers in order for a call to be successfully placed between them. At the originating end, a coding profile is assigned to a destination telephone number. When a call request for a particular destination is received from the telephony interface at the originating end, the parameters from the associated coding profile are negotiated with the remote peer via Dexter's proprietary signaling message elements. At the remote end, a coding profile will have been associated with the telephony destination through prior configuration.

Common elements from the originating side's coding profile and the destination side's coding profile are then negotiated and converged upon (via signaling) to create the set of parameters used to configure the associated DSP voice channels at both ends. Once this process is completed, the voice call is considered active.

Dial Plan Configuration

In addition to physical resource configuration (PBX mode, FXO, FXS, etc.), a dial plan that specifies how to route calls between Dexter peers is required. Dexter maintains its own dial plan that contains the following information:

- Dialled digit timeouts and termination sequences
- Narrowband hunt group definitions
- Broadband hunt group definitions
- Forwarding criteria.

SNMP

Standard MIB support for Dexter includes:

- RFC 1406 Standard T1/E1 MIB
- Supplemental MIB supporting ANSI T1.231.

Additionally, Dexter is configured with its Enterprise MIB structure to facilitate the reporting of non-standard object elements such as ISDN PRI information.

Network Management Processing

This subsystem provides the facility to control and configure Dexter's different subsystems.

Overview

Briefly, the Network Management Subsystem comprises four main components that enable a network operator to configure, control, report, and perform diagnostics upon Dexter. These elements are:

- Configuration Management
- Connection Management
- Fault Management
- Performance Management.

Configuration Management

This component provides functions to configure all aspects of Dexter's physical interfaces, signaling protocol parameters, and call control parameters. From a management perspective, this involves the following entities:

- General node configuration
- E1/T1 port and subchannels
- BRI-ISDN, 10BaseT, V.35 , and RS-232C ports
- ATM and IMA ports
- Narrowband signalling
- Inter-Dexter communications
- Voice coding profiles
- Routing, narrowband, and broadband addressing tables
- OAM segmentation end points table
- Frame Relay and IP interworking tables
- CES configuration.

Connection Management

Connection Management is a set of functions that is used to track the various call or connection oriented entities and configuration of PVCs, including applications they support. From a node management perspective, this involves describing the details of:

- Active call connections between narrowband and broadband resources
- Active broadband connections for the total system
- PVCs created for the broadband entities

- PVCs created for the narrowband entities
- Call history information.

Fault Management

Fault Management is a set of functions that enable the detection, isolation, and correction of abnormal operation of the telecommunications parts of the network and its environment. From a node perspective, this tracks the following entities:

- Physical facility and port failures
- Call control failures
- ATM OAM cell loopback tests
- Sundry fault management and vendor-specific diagnostics.

Performance Management

Performance Management provides functions to evaluate and report upon the behavior of telecommunication/data equipment and the effectiveness of the overall network or network element. From a node management perspective, this involves general performance, traffic, and data collection routines against the following entities:

- Physical layer performance monitoring of all ports
- Cell level performance monitoring
- ATM layer protocol and performance monitoring.

This page left intentionally blank

Chapter 5

Standards Compliance

This section lists the standards and compliance specifications relevant to Dexter.

ANSI Documents

(1) T1.CBR-199X Draft - Broadband ISDN - ATM Adaptation Layer for Constant Bit Rate Services, Functionality and Specification, November 1992.
(2) T1.102-1993, Digital Hierarchy, Electrical Interfaces, December 1993.
(3) T1.107-1995, Digital Hierarchy, Formats Specifications, 1995.
(4) T1.231-1993, Digital Hierarchy, Layer 1 In-Service Digital Transmission Performance Monitoring, September 1993.
(5) T1.403-1995, Carrier-to-Customer Installation, DS1 Metallic Interface, March 1995.
(6) T1.408-1990, Integrated Services Digital Network (ISDN) Primary Rate – Customer Installation Metallic Interfaces Layer 1 Specification, September 1990.
(7) T1.606, T1.606a, T1.606b Frame Relay Bearer Service, Architectural Framework and Service Description, ANSI, 1990.
(8) T1.646-1995, Broadband ISDN, Physical Layer Specifications for User-Network Interfaces Including DS1/ATM, 1995.
(9) EIA/T1A-547, Network Channel Terminal Equipment for DS1 Service, March 1989.

ATM/Frame Relay Forum Documents

(10) AF-VTOA-0078.000, Circuit Emulation Service Interoperability Specification, Version 2.0, January 1997.
(11) The ATM Forum, af-vtoa-0089.000, "Voice and Telephony Over ATM - ATM Trunking using AAL1 for Narrowband Services Version 1.0", July 1997.
(12) The ATM Forum, af-phy-0086.000, "Inverse Multiplexing for ATM (IMA) Specification, Version 1.0, July 1997.
(13) The ATM Forum, af-vtoa-0113.000, "ATM Trunking using AAL2 for Narrowband Services", Version 1.0, February 1999.
(14) UTOPIA, An ATM-PHY Interface Specification, Level 2, Version 0.95, June 1995.
(15) ATM User-Network-Interface Specification, Version 3.1, September 1994, ATM Forum.
(16) UTOPIA, An ATM-PHY Interface Specification, Level 2, Version 0.95, June 1995, ATM Forum.
(17) Network Working Group, RFC 1483, "Multiprotocol Encapsulation over ATM Adaptation Layer 5".

(18) FRF.1.1, User-to-Network Implementation Agreement.
(19) FRF.3.1, Frame Relay Forum Multiprotocol Over Frame Relay.
(20) Frame Relay/ATM PVC Network Interworking Implementation Agreement, Document Number FRF.5, December 20, 1994.
(21) Frame Relay Forum. Frame Relay/ATM PVC Service Interworking Implementation Agreement, Document Number FRF.8, April 15, 1995.

IETF

(22) RFC 1483 Multiprotocol Encapsulation Over AAL5, July 1993.
(23) RFC 1490 Multiprotocol Interconnect Over Frame Relay, July 1993.
(24) RFC1577 Classical IP and ARP over ATM, January 1994.

ITU Documents

(25) ITU-T Recommendation G.168, Digital Network Echo Cancellers, April 1997.
(26) Draft new ITU-T Recommendation I.363.2, B-ISDN ATM Adaptation Layer Type 2 Specification, February 1997.
(27) ITU-T Recommendation I.362 B-ISDN ATM Adaptation Layer(AAL) Functional Description.
(28) ITU-T Recommendation I.363 B-ISDN ATM Adaptation Layer(AAL) Description.
(29) Recommendation G.703, Physical/Electrical Characteristics of Hierarchical Digital Interfaces, 1991.
(30) Recommendation G.704, Synchronous Frame Structures Used at Primary and Secondary Hierarchical Levels, 1991.
(31) Recommendation G.706, Frame Alignment and Cyclic Redundancy Check (CRC) Procedures Relating to Basic Frame Structures Defined in Recommendation G.704, 1991.
(32) Recommendation G.804, ATM Cell Mapping into Plesiochronous Digital Hierarchy (PDH), January 1993.
(33) Recommendation G.823, The Control of Jitter and Wander Within Digital Networks Which are Based on the 2048 kbit/s Hierarchy, 1993.
(34) Recommendation G.826, Error Performance Parameters and Objectives for International, Constant Bit Rate Digital Paths at or above the Primary Rate, 1993.
(35) Recommendation G.832, Transport of SDH Elements on PDH Networks: Frame and Multiplexing Structures, 1993.
(36) Recommendation I.233.1, Framework for providing additional packet mode bearer services, ITU-T, 1988.
(37) Recommendation I.370, Congestion management for the ISDN Frame Relaying bearer service, ITU-T, 1988.
(38) Recommendation I.431, Integrated Services Digital Network (ISDN) User-Network Interface, Primary Rate UNI Layer 1 Specification, March 1993.
(39) Recommendation I.432, Broadband Integrated Services Digital Network (B-ISDN) User-Network Interface, Physical Layer Specification, March 1993.
(40) Recommendation I.610, Broadband Integrated Services Digital Network (B-ISDN) Operation and Maintenance, Principles and Functions, March 1993.
(41) Recommendation Q.922 ISDN Data Link Layer Specification for Frame Mode

Bearer Services, 1992.

(42) ITU-T Recommendation Q.931, DSS1 - ISDN User-Network interface layer 3 specifications for basic call control.

(43) Recommendation Q.933, Digital Subscriber Signaling System No. (DSS 1), Signaling For Frame Mode Basic Call Control, ITU-T, 1993.

Other Related Documents

(44) EN50082-1 "Electromagnetic compatibility, Generic immunity standard, Part 1: Residential, commercial and light industry". EN 50082-1:1997 (or BS EN 50082-1:1998).

(45) ENV 50204 "Radiated electromagnetic field from digital radio telephones - Immunity test". ENV 50204:1995.

(46) IEC 61000-4-2 "Electromagnetic compatibility (EMC), Part 4-2: Testing and measurement techniques, Electrostatic discharge immunity test". IEC 61000-4-2 Consol. Ed. 1.1 (incl. am1), 1999-05.

(47) IEC 61000-4-3 "Electromagnetic compatibility (EMC), Part 4-3: Testing and measurement techniques, Radiated, radio-frequency, electromagnetic field immunity test". IEC 61000-4-3 - Consol. Ed. 1.1 (incl. am1), 1998-11.

(48) IEC 61000-4-4 "Electromagnetic compatibility (EMC), Part 4: Testing and measurement techniques, Section 4: Electrical fast transient/burst immunity test. Basic EMC Publication". IEC 61000-4-4 - Ed. 1.0, 1995-01.

(49) IEC 61000-4-5 "Electromagnetic compatibility (EMC), Part 4: Testing and measurement techniques, Section 5: Surge immunity test". IEC 61000-4-5 - Ed. 1.0, 1995-02.

(50) IEC 61000-4-6 "Electromagnetic compatibility (EMC), Part 4: Testing and measurement techniques, Section 6: Immunity to conducted disturbances, induced by radio-frequency fields". IEC 61000-4-6 - Ed. 1.0, 1996-04.

(51) IEC 61000-4-8 "Electromagnetic compatibility (EMC), Part 4: Testing and measurement techniques, Section 8: Power frequency magnetic field immunity test. Basic EMC Publication". IEC 61000-4-8 - Ed. 1.0, 1993-06.

(52) IEC 61000-4-11 "Electromagnetic compatibility (EMC), Part 4: Testing and measuring techniques, Section 11: Voltage dips, short interruptions and voltage variations immunity tests". IEC 61000-4-11 - Ed. 1.0, 1994-06.

(53) EN50081-1 "Electromagnetic compatibility, Generic emission standard, Part 1: Residential, commercial and light industry". EN 50081-1:1992.

(54) FCC Part 15 "RADIO FREQUENCY DEVICES". Downloaded October 1998. Federal Communications Commission, USA.

(55) EN55022 "Information technology equipment, Radio disturbance characteristics, Limits and methods of measurement", CISPR 22 - Ed. 3.0 - Bilingual, 1997-11, or EN 55022:1998.

(56) EN 55014-1 "Electromagnetic compatibility, Requirements for household appliances, electric tools and similar apparatus, Part 1: Emission, Product family standard". EN 55014-1:1993/A2:1999.

(57) EN 61000-3-2 "Electromagnetic compatibility (EMC), Part 3-2: Limits - Limits for harmonic current emissions (equipment input current up to and including 16A per

phase)". EN 61000-3-2:1995/A2:1998.

(58) EN 61000-3-3 "Electromagnetic compatibility (EMC), Part 3: Limits - Section 3: Limitation of voltage fluctuations and flicker in low-voltage supply systems for equipment with rated current up to 16 A". EN 61000-3-3:1995.

(59) IEC60950 "Safety of information technology equipment." IEC 60950 (1999-04) (Ed.3).

(60) EN60950 "Safety of information technology equipment." EN 60950:1992/A4:1997.

(61) UL 1950 "Standard For Safety For Information Technology Equipment", UL 1950_3 third edition 1995.

(62) UL 1459 "Standard For Safety For Telephone Equipment", UL 1459 third edition 1995.

(63) EN 41003 "Particular safety requirements for equipment to be connected to telecommunication networks", EN 41003:1998.

Product Specification

Mako

1. Introduction

1.1. Scope

Mako is a Mariner Networks proprietary technology that performs frame to cell conversion and data forwarding in hardware. It consists of a cell switching fabric with Segmentation and Reassembly at the edge. It contains the necessary logic and tables to translate between ATM VC's and Frame Relay DLCI's. It also contains scheduling ability to support ATM and Frame Relay Quality of Service (QoS).

This document describes the function and architecture of an implementation of Mako technology for the Dexter 30xx-series of switches that will increase throughput to wire speed. The objective is to convert and forward up to a maximum of 1024 data streams entering through an array of 8 Frame Relay ports, at up to 2.048 Mbps per port, to correlated ATM connections passing through a single UTOPIA level 2 interface while concurrently converting and forwarding up to a maximum of 1024 incoming ATM connections to an array of 8 Frame Relay ports, at up to 2.048 Mbps per port.

In summary, this Mako implementation is expected to convert and forward up to 1024 full duplex connections between Frame Relay and ATM at an aggregated throughput of up to 32 Mbps.

1.2. Document Overview

This document is intended to be a self-contained description of the requirements for the Mako. It is intended to serve as the controlling technical document for the engineering development effort.

1.3. References

(1) ATM User-Network-Interface Specification, Version 3.1, September 1994, ATM Forum.

(2) UTOPIA, An ATM-PHY Interface Specification, Level 2, Version 0.95, June 1995, ATM Forum.

(3) Frame Relay/ATM PVC Network Interworking Implementation Agreement, Document Number FRF.5, Dec 20, 1994, Frame Relay Forum.Frame Relay/ATM PVC Service Interworking Implementation Agreement, Document Number FRF.8, Apr 15, 1995, Frame Relay Forum.

(4) Voice over Frame Relay Implementation Agreement, Document Number FRF.11, Dec 1998, Frame Relay Forum.

(5) Frame Relay Fragmentation Implementation Agreement, Document Number FRF.12, Frame Relay Forum.

2. Features

Mako will support the following:

- Utopia level 2 interface for ATM cell ingress and egress.
- Processor interface for both configuration/control and data.
- 1024 virtual connections.
- AAL-5 segmentation and reassembly.
- 32 Mbps throughput.
- ATM QoS support per VC: CBR, VBRrt, VBRnrt, UBR.
- Per port pacing.

- Frame Relay QoS support per DLCI: CIR.

4. Functional Description

Figure 4.1 Mako Block Diagram

Inside the Makodata is held as ATM cells in a common buffer memory. Each cell is placed in a buffer upon ingress and remains in that same buffer until egress. Inside Mako, cells are virtually forwarded by physically passing only the buffer number (BN).

Frames enter the TDM Ports interface directly from a TDM highway. The Tdm Res block translates the incoming port and DLCI numbers to an internal Ci (Connection Index), via a binary search table lookup in the TRT (TDM Resolution Table).

The Seg block converts the frame to an AAL-5 cell stream and presents the AAL-5 cells to the PointerSwitch for forwarding. Depending on information in the XT (Switch Table), cells are forwarded either back out to TDM port, to the CPU, to an ATM Cell Port or to the HoldQueue.
Cells belonging to outgoing Frames enter the Tdm Reas block from the PointerSwitch. Based on per Ci information in the TXT (Tdm Translation Table), Tdm Xlt adds appropriate headers and trailers and inserts the correct DLCI and port numbers. Tdm Xlt then passes the Frames out through the Tdm I/F.

Incoming ATM cells streams enter through the Utopia port where they are presented to the CellRes block. CellRes translates the VPI, VCI and PTI fields to a Ci, via a binary search table lookup in the URT (Utopia Resolution Table).

CellRes presents incoming cells to the PointerSwitch. The PointerSwitch virtually switches cells to their next destination. Depending on information in the XT (Switch Table), cells are forwarded either back out to an ATM cell port, to the CPU, to a TDM Port or to the HoldQueue.

Outgoing ATM Cells enter the CellXlt block from the PointerSwitch. CellXlt uses per-Ci information in the UXT (Utopia Translation Table) to assign the appropriate VPI and VCI to outgoing cells.

CellXlt then passes the Cells out through the Cell I/F.

When a cell stream requires buffering and retransmission with controlled QoS, its cells are switched to the CellSched block. CellSched holds queued cells until QoS information in the ST (Schedule Table) indicates they should be forwarded back out to the outside world. A proprietary register array, the BT (Bubble Table) is used to determine cell order.

The CellSched block is a multi-service cell scheduler. It receives per-Ci QoS requirements from the ST (Scheduler Table). Based on information in the ST and conditional on Cell availability information from the HoldQueue block, it sequences cell forwarding requests per-Ci compliant with CBR, VBRrt, VBRnrt and UBR standards.

5. Theory of Operation

Mako is a cell based core switch with cell to frame and cell to packet conversion at the edge. The Mako

implementation supports Frame but not Packet ports so it only has cell to frame conversion and no cell to packet conversion.

5.1. Cell Buffering and Queuing

Data from any port either arrives as cells or is converted to cells. Each cell is assigned a unique Buffer Number (BN) in the range of 1 to MaxBN-1. In Mako, MaxBN is expected to be about 1800. The exact value won't be determined until late in the design phase. Each BN corresponds to 52 byte cell buffer (BPld) and a block of pointers (BPtr). Incoming cell data is stored in the BPld associated with its BN. Values are adjusted in the cell's BPtr which position the cell's BN in a threaded queue for processing. There are also a BPtr and BPld associated with BN = 0. BN = 0 is a special case associated with ATM Idle Cells.

BN's are always ordered in threaded queues. A queue of BPtr's is used to link BN's within queues. Each BPtr contains the following fields:

BPtrNxt contains the next BN in the threaded queue.
BptrClSel contains a one bit selector bit for PointerSwitch clients with dual client ID's.

The BptrClSel field is not related to queueing but is packaged within BPtr for convenience. It's function will be explained later.

5.2. Connection Queuing

Mako tracks connections by unique Connection Indexes (Ci's) in the range of 0 to MaxCi - 1. In Mako, MaxCi is 1024. There are a number of lists that are indexed by Ci which are used to control processing and forwarding of cells on a per-connection basis.

After initialization, there are no cells in the system. All BN's are ordered in the Free Queue (FreeQ). There is a list of empty pointers (CiPtr's), indexed by Connection Index (Ci). As cells enter Mako and are processed and forwarded, these pointers will become BN queue pointers. BN's will be moved from FreeQ to the CiPtr's and back to FreeQ. Each CiPtr contains the following elements:

CiIn contains the BN of the most recent cell from an input port.
CiSch contains the BN of the next cell to be processed by the CellSched block.
CiPktFrm contains the BN of the next cell to be processed by FrmXlt.
CiOut contains the BN of the next cell to be sent out to an output port.

CiIn is also the beginning of the queue and CiOut is the end.

A general implementation of Mako would include another element in CiPtr, named CiL3, which would contain the BN of the next cell to be processed by the Layer3 block. However, the Layer-3 processing block is presently not included and neither is its corresponding pointer element.

5.3. Cell Input

ATM cells ingressing through the Cell Interface are examined by the CellRes block. The VPI, VCI and PTImsb (most significant bit of the PTI field) are compared against entries in the Utopia Resolution Table

(URT) for a match. Each entry in the URT contains the following fields:

UrT(Us)->Key (bits 0 - 31 of 1st word)
UrT(Us)->Ci (bits 0 - 12 of 2nd word)
UrT(Us)->unused (bits 13 - 31 of 2nd word)

UrT(Us)->Key contains the following sub fields:
Key->PtiMsb (bit 0) = ATM hdr TPI field msb
Key->Vci (bits 1 - 16) = ATM hdr VCI field
Key->Vpi (bits 17 - 24) = ATM hdr VPI field
Key->Phy (bits 25 - 29) = Utopia PHY number
Key->Ut (bits 30 - 31) = Utopia number
Entries in UrT are ordered by the value in Key.

The URT is ordered by bhe key field and searched by a binary search algorithm. If a match is found for the key of the incoming cell, the Ci is read and the cell is passed on to the PointerSwitch. If no match is found, the UnkCell counter is incremented and the cell is discarded.

5.4. Cell Output

The CellXlt block receives Ci's from the PointerSwitch block. On receipt of a Ci, CellXlt adds the Ci into the CellOut FIFO. Each element in the CellOut FIFO contains one field:
CxCi contains the Ci to which the cell applies.

If the CellOut FIFO is not empty, CellXlt removes the first entry. CellXlt indexes the CiPtr table by the Ci removed from the CellOut FIFO and retrieves the BN from CiPtr->CiOut. It then indexes the BPld table by this BN to find the header and payload of the cell to be output. CellXlt then uses the Ci from the CellOut FIFO to index the Utopia Translation Table (UXT). Each entry in the UXT contains the following fields:

Uxt->Vpi (bits 0 - 15 of 1st wd) = Outgoing VCI or 0xFFFF if don't xlate
Uxt->Vci (bits 16 - 31 of 1st wd) = Outgoing VPI or 0xFF if don't xlate
Uxt->Msk (bits 0 - 31 of 2nd wd) = IP mask if is routed Ci

Uxt->RtCi (bits 0 - 13 of 3rd wd) = Ci if being routed
Uxt->Fwd (bit 14 of 3rd wd) = 1 if currently forwarding AAL-5 cells
Uxt->Rtd (bit 15 of 3rd wd) = 1 if is routed Ci

CellXlt conditionally replaces the VPI and VCI fields in outgoing cell header and sends the cell to the Cell Interface. CellXlt then calls RlsBf, in the CellPtrs block, to release the BN back to the Free Queue.

CellXlt checks the CellOut FIFO. If the FIFO is not empty, another Ci is and processed.

5.5. Cell Routing

PointerSwitch clients in the Mako implementation consist of ATM, TDM, CPU and Scheduler. All but the last have a single unique ClId (Client ID). The Scheduler has two ClId's so it can differentiate forward and backward direction with regard to the Ci as viewed from the external ports of the Ci. For example, consider a Ci that is defined between an ATM port and a TDM port. When the Scheduler sends a cell for this Ci to the PointerSwitch, there must be a way to know whether it is a cell that came from the ATM port and now

must be sent to the TDM port or vice versa. The duality of CIId's accomplishes this.

In Mako, CIId assignments are:

CIId = 0 for CPU port
CIId = 1 for ATM Cell Port
CIId = 2 for TDM Port
CIId = 3 or 4 for Scheduler

As cells pass through Mako, they are processed by various logic blocks. The order in which they are processed, i.e. the routing order from block to block, is determined by information in the XT (Switching Table). The XT is indexed by Ci. Each XT table element consists of two fields, the first bit, 15, being the scheduler bit saying whether this Ci is to pass through the scheduler; bits 14 - 0 hold the destination port, indicating which port this Ci is to be routed to. The following examples illustrate the function of the XT.

Consider for example a Ci numbered 200 which is to be a simple VC from ATM to TDM port with no QoS considerations. The fields in XT[200] are:

XT[200] = 0, 65 - 1022   where this is any number in the range of TDM ports on Mako

Consider another example of a Ci numbered 123 between ATM and TDM in which data in both directions is to be buffered and paced out to guarantee QoS.

XT[123] = 1, 65 - 1022   where the 1 indicates that this Ci must pass through the scheduler

Consider an example of a Ci numbered 753 between ATM and TDM in which Frame Relay is to be buffered and paced out to ATM but ATM is to be forwarded directly out to TDM. This would be typical of local Frame Relay users connected to a policed public network through ATM.

XT[753] = 1, 65 - 1022

Consider a final example of OAM cells being exchanged between ATM and the CPU.

XT[567] = 0, 1023   where 1023 is the port number of the local port interface to the CPU.

5.6. Cell Scheduling

Note from WPB: Section 5.6 is undergoing extensive revisions and can be considered a work in progress.

5.6.1. Introduction

Cell scheduling (more properly cell sequencing) is performed by the Cell Scheduler (CS). The Cell Scheduler is the distinguishing core of Mako technology. If in the final design of the Mako, the Cell Scheduler becomes the pacing bottleneck, then the balance of the design has reached their practical limits. The CS will have some variations possible but this spec is intended for use in the Dexter 3000. The variations may be a

trimming for FRAIM and Dexter 2210 use, with expansion for Dexter 4000 use. All of the needed setup calculations and initial setup will need to be performed by the CPU thus releasing the CS to do its job without impact on the switching bandwidth. The CS will be able to perform pacing (policing) on a per-port basis. It will insert idle cells as required. One important provision of the scheduler that should be noted is that ports will not intermingle direct switched cell streams with scheduled cell streams. If a single VC is scheduled (retimed) for a port all VC's for that port must be scheduled.

5.6.2. Block Structure of Cell Scheduler and RAM Allocation

The following diagram is a block representation of the CS.

Drawing to be inserted when completed.

Figure 5.6.1 Cell Scheduler Structure

The following tables show the Cell Scheduler use of internal and external RAM.

Number of Internal RAM bits (Bytes)	Use of RAM block
16,384 bits (2kB)	Ci status register table (8k x 2 bits)
17,408 bits (2,176B)	Port Status Register (1k x 17 bits)
11,264 bits (1408B)	Internal Port Table Entries (32 x 352 bits)
131,072 bits (16kB)	Internal Bubble Tables (BT) (32 bits x 4096, 128 BT pages)
128 bits (16B)	Internal Bubble Table Entry Mapping (1 bit x 128)
2048 bits (0.25kB)	External Bubble Table Mapping (1 bit x 2048)
FREE 34,688 bits	
TOTAL 212,992 bits (26kB)	

Table 5.6.1 Cell Scheduler Internal RAM Allocation

Number of External RAM bits (Bytes)	Use of RAM block
1,572,864 bits (192kB)	Scheduler Table (ST) (8k x 24 bytes)
360,448 bits (44kB)	Port Table (PT) (1k x 44 bytes)
2,097,152 bits (256kB)	External Bubble Tables (BT) (32 bits x 65,536 each, 2048 BT pages)
TOTAL 4,030,464 bits (492kB)	

Table 5.6.2 Cell Scheduler Internal RAM Allocation

5.6.3. Cell Scheduling Algorithm Flow

The Cell Scheduler Flow and Process diagram is kept in a separate Visio file named CSflow.vsd. The detailed discussion that follows refers to this flow diagram. It is impractical due to its size and complexity to convert to Word format. It is included in this document by reference. There are parallel processes performed for port sequencing and Ci activation. In Section 5.6.4 the Ci Activation Process is detailed and discussed. In Section 5.6.5 Pre-Scheduling Port Process is detailed and discussed. Finally, in Section 5.6.6 Post-Scheduling Port Process is detailed and discussed. These three processes run independently and in parallel of the Scheduler thus freeing the scheduler for scheduling cells for transmission. This should enhance utilization of available bandwidth.

This flow can be viewed most conveniently as consisting of 13 sub-flows. Many functions in the sub-flows are similar but will be performed sequentially and not in parallel. The most important part of the scheduler is the Bubbler. This flow is designed to make the most efficient use of the Bubbler. Table 5.6.3 shows the steps that are associated with each sub-flow.

Sub-flow function	Steps of sub-flow
Ci Activation	Steps 1 to 7
Port Overhead	Steps 8 to 13
QoS PCBR	Steps 14 to 16
QoS CBR	Steps 17 to 26
QoS VBRrt	Steps 27 to 39
QoS VBRnrt	Steps 40 to 52
QoS ABR	Steps 53 to 65
QoS QFC	Steps 66 to 82
QoS ThVBRrt	Steps 83 to 92
QoS ThVBRnrt	Steps 93 to 102
QoS ThABR	Steps 103 to 112
QoS UBR	Steps 113 to 122
QoS Idle Cell Fill	Step 123

Table 5.6.3 Sub-flow Divisions in Scheduler Flow

Scheduler Flow is kept in separate Visio file named CSflow.vsd.
Flow 5.6.1

5.6.3.1. Step 1 Detailed Description

This is a simple test to see if there is a Ci requiring activation. To minimize delays in activating Ci's this test is performed every loop through the CS. This can be especially critical on data streams that become data starved with every cell transmitted. The Ci Activation queue (CiAq) is discussed in precise detail in section 5.6.4 on Ci Activation. This step determines if there is an entry in the CiAq. By convention an entry of Ci equal to zero will be the same as no entry. If the current entry is equal to zero then no increment of the pointer will take place. If there is none, control is passed to the port sequencing part of the flow, Step 8. Otherwise control continues on to Step 2.

5.6.3.2. Step 2 Detailed Description

In this step the number of the Ci to be activated is read from the CiAq and the read pointer incremented. Data on Ci's are contained in 2 places. The first is an internal status register table. This table arranged by Ci number is two bits representing the status of the Ci. Table 5.6.4 shows the structure of these registers. Bit 0 shows whether the Ci is active. The Ci Activation process will set bit 0 when placing a Ci on the CiAq. If another cell for the same Ci were to follow, only a single activation will occur. When this process sends the last cell in a Ci thread it will clear this bit. QFC Ci's use bit 1. This register table will be stored in internal RAM and will use 16kb (16,384 bits) of RAM. That's 2 bits for each of 8k Ci's. Bit 1 will indicate that the Ci has been blocked because of lack of credit downstream. Bit 1 will stop incoming cells on these Ci's from falsely triggering a reactivation of these channels.

Bit Number	
1	**0**
QFC-Blocked	**Ci Active**

Table 5.6.4 Internal Ci Status Register Structure

The second place for Ci settings is in the Scheduler Table (ST). Table 5.6.5 shows the general structure of an entry in the ST. The first double word functions the same for all QoSs. Bits 0-2 are the 3 fractional bits for the Interval Count. Bits 3-16 are the integer portion of the Interval Count. These give us a 14.3 bit resolution. The utilization of available bandwidth may be expressed as 1/IC. If IC = 1.0 then 100% of the bandwidth is used. If IC = b1.001 (decimal 1.125) then 88.88...% of the bandwidth is used. At the other end if Ci = b11111111111111.111 (32767.875) then 0.00306% of the bandwidth is used. The value for this number depends on the QoS selected. It is calculated by

software and is critical to the operation of the CS. It should be calculated as close as possible then rounded down thus rounding up the bandwidth used and ensuring contract requirements are met. There are 1k possible ports so bits 17 to 26 are the 10 needed bits for the port number of the Ci. Bits 27-29 show the Quality of Service (QoS) for this Ci. Table 5.6.6 shows the various QoSs used by Mako. Please note that while there are 11 listed QoSs, only 7 values are used for the QoS value. Only the Port Status Register and Bubble Tables use the throttled QoSs. Bits 30 and 31 are unused at the present time. The ST is kept in external RAM and is indexed by Ci number. There are 8k possible Ci's and each entry in the ST uses 24 bytes, therefore the ST will use 192kB of external RAM.

BIT NUMBER
31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0

Unused	QoS	Port Number
Integer Portion of Interval Count		Fractional Portion of Interval Count
Use of these registers varies by QoS detailed in section on each QoS.		
Use of these registers varies by QoS detailed in section on each QoS.		

Table 5.6.5 General ST Register Structure

QoS Priority Name		QoS Value (Binary, Decimal)	QoS
1	000, 0	Pacing Constant Bit Rate (PCBR, forced idles)	
2	001, 1	Constant Bit Rate (CBR)	
3	010, 2	Variable Bit Rate, real time (VBRrt)	
4	011, 3	Variable Bit Rate, non-real time (VBRnrt)	
5	100, 4	Available Bit Rate (ABR)	
6	101, 5	Quantum Flow Control (QFC)	
7	010, 2-A	Throttled VBRrt Ci's (ThVBRrt)	
8	011, 3-A	Throttled VBRnrt Ci's (ThVBRnrt)	
9	100, 4-A	Throttled ABR Ci's (ThABR)	
10	110, 6	Unspecified Bit Rate (UBR)	
11	Implied	Idle Cell Filler, added by state machine	

Table 5.6.6 Available Qualities of Service (QoSs)

There are 3 types of QoSs listed that are not industry standard and are unique to Mako. The first is the Pacing Constant Bit Rate (PCBR). A PCBR Ci is used to force idle cells on an active port to limit the use of available bandwidth. An example of this could be when the physical connection is a T1 1.554Mb connection and the user is only paying for half the bandwidth. The PCBR would force idle cells every other cell thus pacing the port to 50%. These Ci's will set up like any others except they will be activated whenever the port becomes active and not through the CiAq. If there are no active Ci's,

thus rendering the port inactive, then the physical layer will fill with all idle cells by default and PCBR is not needed. Sending PCBR cells and idle fill cells advance the Port Increment Counter and advance the conditions for unthrottling. The PCBR is a special case in that a combination of up to 3 Ci's may be used to get the finest possible resolution on the pacing value. This cannot be done with other Ci's, as only a single cell stream is manageable and available, with PCBR the cell stream is all idle cells.

The second unique case QoS is for throttled Ci's. These Ci's have sent their guaranteed number of cells per time measurement unit. These are VBR and ABR Ci's. These Ci's have been moved to lower priority thus giving CBR and unthrottled Ci's more of the available bandwidth (higher priority). Under QFC there is a credit/debit system that when exhausted that causes the Ci to be blocked (deactivated) rather than throttled. The time measurement unit will be set between 1/8 and 1/4 of a second and the appropriate number of guaranteed cells set. Every time a cell is sent this number is decremented in the Ci's register. When zero is reached the Ci will be removed from the unthrottled QoS BT and placed in the throttled QoS BT. Cells are then sent over the switch at the new lower priority. When the time measure counter resets to zero these Ci's will be unthrottled and placed back in the correct QoS to schedule cells once again. This unthrottling is explained in Step 13.

The third and final special case QoS is the implied Idle Cell Filler. This QoS has no assigned Ci number. When a port is active and there are no other cells ready to be sent, then this QoS inserts an idle cell to maintain the bandwidth integrity of the port. This is especially important on ports with only active CBR Ci's. If there were no idle cell filler then the port would over schedule the CBR Ci's.

5.6.3.3. Step 3 Detailed Description

This step uses the port number of the activating Ci and checks to see if the port is active by checking the port active bit in the Port Status Register. If the port is active Step 4 is bypassed and control transferred to Step 5. If the port is not active then control is passed to Step 4.

5.6.3.4. Step 4 Detailed Description

This step does not activate the Ci but rather prepares a given port for Ci activation. For a port to be inactive it means that there was no data available for any of the port's Ci's.

This step will activate the port by setting the port active flag and activating any PCBR Ci's that may be associated with the port. In internal RAM there shall be a Port Status Register. This register has 16 bits for each port. Since there 1k ports this will use 16kb (16,384 bits) of internal RAM. The structure of these status registers is shown in Table 5.6.7. Bit 15 is set for an active port. Bits 5 to 14 are set to show which QoSs have active Ci's. Bit 4 is set when the PT entry has been transferred to internal RAM. There will be 16 available spaces in internal RAM for PT entries, so bits 0 through 3 (4 bits) will be used for addressing. Each entry uses 352 bits, therefore all 16 stored entries will use 5,632 bits of internal RAM. All entries will be also be stored in external RAM. Each entry will use 44 bytes and there are 1k maximum ports, therefore the PT will use 44k bytes of the external RAM. Since a port is being activated here there will be no pre-loading of the port entry but rather a read from external RAM. In this step the Port Active bit is set. The Port Interval Counter will not be reset. This is important to maintain bandwidth integrity for bursty traffic. When a port is first setup by software it is important to set the PIC to zero giving the port a uniform place to begin processing cells. Any PCBR Ci's are activated. They should activated with target times equal to the PIC. This forces the PCBR Ci's to start sending idle cells first before other Ci's thus enforcing pacing requirements from the beginning. The reason the PIC is more than 14 bits is to allow for scaling on the time measurement unit on ports of various speeds. This scaling factor is in terms of bits from 14 to get a time measurement unit between 1/8 and 1/4 of a second. Table 5.6.8 shows the structure of an entry in the PT. Please note the maximum number of active Ci's for a single port's QoS is 1,023 (1k - 1) or 10 bits. It will place the appropriate addresses for first Ci of a QoS reading. When the CS is checking for ready cells this base address will have the target time for the highest priority Ci for a particular Ci and can be used for a fast check for ready to send status. The PT entry should be retained in internal registers for the next step.

BIT NUMBER
16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
Port Active PCBR QoS Active CBR QoS Active VBRrt QoS Active
VBRnrt QoS Active ABR QoS Active QFC QoS Active
Throttled VBRrt QoS Active Throttled VBRnrt QoS Active
Throttled ABR QoS Active UBR QoS Active Internal RAM Used
Port Table Internal RAM location

Table 5.6.7 Port Status Register

BIT NUMBER

31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16
15 14 13 12 11 10 9 8 7 6 5 4 3 2 10

Double Word Number
0 Signed Scaling Factor for Time Measure Unit
Port Interval Counter (PIC)
1 Number of PCBR Ci's Located in Internal RAM
Base Address of this PCBR Queue
2 Number of CBR Ci's Base Address of this CBR Queue
3 Number of VBRrt Ci's
Base Address of this VBRrt Queue
4 Number of VBRnrt Ci's
Base Address of this VBRnrt Queue
5 Number of ABR Ci's Base Address of this ABR Queue
6 Number of QFC Ci's Base Address of this QFC Queue
7 Number of ThVBRrt Ci's
Base Address of this ThVBRrt Queue
8 Number of ThVBRnrt Ci's
Base Address of this ThVBRnrt Queue
9 Number of ThABR Ci's
Base Address of this ThABR Queue
10 Number of UBR Ci's
Base Address of this UBR Queue

Table 5.6.8 Structure of Port Table Entry

5.6.3.5. Step 5 Detailed Description

This step checks the QoS active bit in the Port Status Register to see if this QoS is set active. If the bit is already set, the flow goes to Step 7. If this bit is set inactive then the flow goes to Step 6.

5.6.3.6. Step 6 Detailed Description

This step sets the QoS active bit active.

5.6.3.7. Step 7 Detailed Description

The QoS BT will be loaded into the Bubbler. This can be either from internal or external RAM depending on the data provided in the PT (there is a possibility that BT has been retained in internal RAM). The Bubbler will then insert the newly activated Ci at the front of the BT, with a Target Time equal to the current PIC for this port. Table 5.6.9 shows the structure of a BT entry. There is one for every active Ci. The Bubbler Processor places it in the proper position. The internal Bubbler will have 32 entries.

Internal RAM will require the use of 1kb (32 entries times 32 bits) for each Bubbler page. The discussions on the Pre and Post Port Processing will show how external RAM and internal RAM are used to coordinate these Bubble sections. The PT entry should be updated (this includes incrementing the QoS active Ci count and adjusting the base of address(es) of the QoS queues as needed.) to show the newly activated Ci and copied to external RAM.

BIT NUMBER

31 30 29 28 27 26 25 24 23 22 21 20 19 18 17
16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
Overflow & Carry **Integer Target Time**
Fractional Target Time **Ci Number**

Table 5.6.9 Structure of Bubble Table (BT) Entry

5.6.3.8. Step 8 Detailed Description

Step 8 gets the next port number to be scheduled from the Port Queue (PQ) and advances the pointer. The PQ is FIFO that gets its entries from the Pre-Scheduling Port Processor. This means that the entries on the PQ have been pre-qualified and are known to active. If the Pre-Scheduling Port Processor has not qualifies any port as active the value here will be zero and control then passes onto Step 1. If the port number is anything but zero control passes to Step 9.

5.6.3.9. Step 9 Detailed Description

The first thing done here is the loading of the Port Register from RAM to the local working register. The Port Status register will tell if this must be done from external RAM or can be done from internal RAM. If a port is active then some type of cell will be sent to the Switch. This means the Port Increment Counter will be advanced. This step does that incrementing in the loaded port register.

5.6.3.10. Step 10 Detailed Description

This step checks to see if the PIC overflowed. An overflow condition would be that bits 0 through 13 would be all set to zero. An overflow would not occur on a newly activated port since the all zeros of such a port would have been incremented to 1. If this condition is true then control passes to Step 10 otherwise the process continues to Step

11.

5.6.3.11. Step 11 Detailed Description

All active Ci's for QoS numbers 1 through 7 are loaded into the Bubbler and the overflow and carry bits are reduced by 01. This will maintain the proper sequence and value for these Ci's now that the PIC has overflowed. Control continues on to Step 12.

5.6.3.12. Step 12 Detailed Description

This step will use the scaling factor in the port entry table to determine if the Time Measure Counter has overflowed. The scaling factor reduces or enlarges the number of bits checked for all 0's. If the scaling factor is set to zero then the Time Measure Counter is equal to 14 bits the same as PIC overflow. If the scaling factor were -5 then only bits 0-8 need to be zero, or 9 bits. Conversely the scaling factor may be as large as +13 meaning that 27 bits (0-26) of the PIC would need to be equal to 0. This would be a very fast port where many cells would need to be sent to meet the 1/8 to 1/4 of a second requirement. Using E1 as a first example: E1 sends 5333.33 cells of data per second. The 14-bit overflow would represent 16,384 (214) cells. This would be over 3 seconds worth of cells. This is too long for effective throttling. In 0.125 seconds E1 would send 666.63 cells and in 0.25 seconds 1333.33 cells. The binary value in this range is 1024 or 210. For E1 the scaling factor would be -4 and a time measure unit of 0.192 seconds. When calculating the maximum number of cells before throttling it should be based on this period of time. Using E4 as an example at the other end: E4 sends 364,583.33 cells per second. The 14-bit overflow would represent 0.04494 seconds. This is far too little for effective throttling. In 0.125 seconds E4 would send 45,572.92 cells and in 0.25 second 91,145.83 cells. The binary number in this range is 216 or 65,536. The E4 scaling factor would then become +2. The time measure period would then be 0.1798 seconds.

5.6.3.13. Step 13 Detailed Description

In this step the Ci's in the three throttled QoS queues will be moved to the appropriate unthrottled QoS queues. The Port Status Register needs to be updated to reflect any changes made.

5.6.3.14. Step 14 Detailed Description

Step 14 checks for a greater than zero value in the number of PCBR Ci's in the Port Table register. If this number is zero then control passes on to Step 16. A number other than zero passes control on to Step 14. This is check for the highest possible priority QoS thus ensuring that PCBR cells are switched in preference to any other QoS.

5.6.3.15. Step 14 Detailed Description

Using the base address in the Port Entry register the Target Time value for the first PCBR Ci is compared to the PIC. If the integral portion is equal or smaller then the Ci is ready to be sent. If true then control passes to Step 18. If false then control moves on to Step 16.

5.6.3.16. Step 15 Detailed Description

A PCBR Ci is passed to the switch in a special manner. It is always an idle cell. The switch receives an idle cell request with the port number instead of the Ci number. The Interval Time for this Ci is added to the Target Time and bubbled back into the BT. Control passes to Step 1.

5.6.3.17. Step 16 Detailed Description

This step checks for active CBR Ci's. This is done by reading the number of CBR Ci's in the Port Table register. If the number is zero control passes to Step 24. If there are active CBR Ci's then control passes to Step 17.

5.6.3.18. Step 17 Detailed Description

Using the base address of the CBR Queue the Target Time is compared with the PIC. If this Ci is ready to send then control passes to Step 18, if not control is passed to Step 24.

5.6.3.19. Step 18 Detailed Description

This step sends this Ci to the Switch. Control passes to Step 19.

5.6.3.20. Step 19 Detailed Description

This step checks to see if this is the last cell on the Ci thread. If it is control passes to Step 20, if not control passes to Step 23.

5.6.3.21. Step 20 Detailed Description

The Ci is removed from the BT for this port. This deactivation is because the Ci has

become data starved. It will be reactivated again by the Port Activation Process when another cell becomes available for switching. Control is passed to Step 21.

5.6.3.22. Step 21 Detailed Description

This step scans for a nonzero value in any of the number of active Ci's for QoSs 1 to 15. If there is a nonzero value then control passes to Step 1. If all values are zero then control passes to Step 22.

5.6.3.23. Step 22 Detailed Description

The Port is deactivated. Control Passes to Step 1.

5.6.3.24. Step 23 Detailed Description

The Interval Time is added to the Target Time and bubbled back into the BT. The Port Entry Table is copied back into RAM.

5.6.3.25. Step 24 Detailed Description

The number of active VBRrt Ci's is read from the Port Table register. If the value is zero control is passed to Step 36, if not control is passed to Step 25.

5.6.3.26. Step 25 Detailed Description

Using the Base Address in the Port Entry register the Target Time for the first VBRrt Ci is checked, if less than the PIC then control is passed to Step 26. If the cell is not ready then control passes to Step 36.

5.6.3.27. Step 26 Detailed Description

The Ci number is sent to the Switch. Control passes to Step 27.

5.6.3.28. Step 27 Detailed Description

This step checks to see if this was the last cell in the Ci thread. If it was then control passes to Step 28, if not then control passes to Step 31.

5.6.3.29. Step 28 Detailed Description

This Ci has become data starved and is removed from the BT. Control passes to Step 29.

5.6.3.30. Step 29 Detailed Description

This step scans for a nonzero value in any of the number of active Ci's for QoSs 1 to 15. If there is a nonzero value then control passes to Step 1. If all values are zero then

control passes to Step 30.

5.6.3.31. Step 30 Detailed Description

The Port is deactivated. Control passes to Step 1.

5.6.3.32. Step 31 Detailed Description

The Ci is still active and is checked to see if the Ci is subject to throttling. If it is then control is passed to Step 32, if not then control is passed to Step 35.

5.6.3.33. Step 32 Detailed Description

Table 5.6.10 shows the structure of the ST entry for a VBRrt entry. There is entry here that is calculated using the PIC and Time Measure Scaling Factor. This is the Time Measure Period Count. This 6-bit count is the bits more significant than the Time Measure Count overflow. When overflow occurs on the Time Measure Period there is no resetting for active unthrottled Ci's. This value accomplishes this needed process. The first check in this step is this value. If it is not equal to the current value from the PIC, then it is set to the PIC value and the number of cells sent is forced to1. This means the Time Measure Period is new and this is the first cell sent by this Ci during it. If the values are equal then the Sent value is incremented by 1. Control is passed to Step 33.

BIT NUMBER
31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
Throttle Throttled QoS
Port Number Integer Portion of Interval Count
Fractional Portion of Interval Count
Maximum Number of Cells per Time Measure Period to Send
Time Measure Period Count
Number of Cells Sent Current Time Measure Period

Table 5.6.10 STE Register Structure for VBRrt

5.6.3.34. Step 33 Detailed Description

The 2 values of the maximum number to be sent is compared with the number sent. If the number sent is equal to or grater than the maximum number control passes to Step 34, if not control passes to Step 35.

5.6.3.35. Step 34 Detailed Description

The Ci is throttled by moving it from the active QoS to the throttled QoS. Control Passes to Step 1.

5.6.3.36. Step 35 Detailed Description

The Interval Time is added to the Target Time and bubble back into the BT. Control passes to Step 1.

5.6.4. Ci Activation Process

The following diagrams the flow of the Ci Activation Process. This process actually qualifies Ci's for activation by the Scheduler. Already active and blocked Ci's are ignored. The sections following give a detailed description of the steps in this process. This flow is also in the Visio file CiAct.vsd.

Flow 5.6.2 Flow for Ci Activation Qualification

5.6.4.1. Step 1 Detailed Description

This is the default idle state for the process. It waits for an alert from the Switch Queue (QSw) or for the CiAq to go from full to non-full. The QSw has a list of Ci's that have had cells switched to the scheduler. If a Ci needs activation it will be place on the CiAq so there is a check to insure there is a place for the entry.

5.6.4.2. Step 2 Detailed Description

This step removes an entry from QSw. It is placed in the register CqCi.

5.6.4.3. Step 3 Detailed Description

The status bits of the Ci are checked to see if it is already active or blocked. If it is either then control transfers to Step 1 else control passes to Step 4.

5.6.4.4. Step 4 Detailed Description

The Ci is added to the CiAq FIFO and the active bit for the Ci is set.

5.7. Frame Segmentation

Frame Relay data coming in through a T1/E1 port is multiplexed by a Line Interface Unit (LIU) onto a TDM highway. 16 T1/E1 ports are allowed. Each LIU can handle 32 channels. This yields 512 channels. Information about each channel is stored in the TST (Time Slot Table). The table has 512 entries with each entry being 76-bits wide. The TST is indexed by the key (Port + Channel + DLCI). Each entry has the following fields:

Status (3 bits)
Ci (or header bytes) (11 bits)
Offset (6 bits)
Frame checksum (16 bits)
AAL5 checksum (32 bits)
Byte fragment (8 bits)

The status field is defined as:

0	idle
1	sync
2	1st byte header
3	2nd byte header
4	3rd byte header
5	4th byte header
6	payload
7	bonding base

(open question: are idles in frames allowed?)

TdmRes block correlates an incoming frame to a defined Ci and passes the Ci to TdmSeg. It does this by searching for the DLCI and source port number of the incoming frame in the TDM Resolution Table (TRT). Each element in the TRT contains the following fields:

TrtDlci contains the DLCI of the Ci.
TrtPort contains the source port of the Ci.
TrtChan contains the source channel of the Ci.
TrtCi contains the Ci.

Elements in the TRT are ordered by the first two fields which serve as the search key. TdmRes performs a binary search in an attempt to match the incoming DLCI and port number to an TRT entry. If the entry is found, the incoming frame is discarded and the UnkFrm counter is incremented. If a match is found, the Ci is passed to the TdmSeg block.

Frame Relay fragmentation and reassembly, recommended but optional in the FRF8 specification, are not supported.

TdmSeg reads the TxCi entry from the entry in the TDM Translation Table (TXT) indexed by Ci. If the mode is FRF5, the following happens: if TxBecn is set, the BECN field is set in the incoming Frame Relay header; depending on TxClp, the CLP bit is copied from the Frame Relay DE field or set to 0 or set to 1 in each cell; the Frame Relay PDU is segmented directly into an AAL-5 cell stream. If the mode is FRF8 transparent. No change is made to the Frame Header and the Frame PDU is segmented directly into an AAL-5 cell stream. If the mode is FRF8 translation then the first six bytes of the Frame Relay header are used in a simple decision tree for form an appropriate header prepended to the AAL-5 PDU.

TrmSeg calls the GetBf routine in the CellPtrs block to get a BN for each cell. TdmSeg reads frame payload from the Frame Interface and stores it in the BPld buffers associated with the BN's.

Only one frame can ingress through the Frame Interface at a time so TrmSeg is not required to interleave between frames.

TrmSeg presents its ClId and a Ci to the PointerSwitch. The PointerSwitch virtually switches cells to their next destination. The next destination may be any client.

5.8. Frame Reassembly

The TrmReas block receives Ci's from the PointerSwitch block. On receipt of a Ci, TrmReas fetches its CiPtr, gets the BN from CiOut and examines the corresponding BPld to see whether it is the last cell of a frame. If it is not the last cell of a frame, TrmReas does nothing more. If it is the last cell of a frame, TrmReas adds Ci and EFCI information into the TrmOut FIFO. Each element in the TrmOut FIFO contains the following fields:

TxCi contains the Ci to which the cell applies.
TxCiBecn indicates whether the EFCI bit was set in the header of the cell in BPld.

If the TrmOut FIFO is not empty, TrmReas removes the first entry. The entry in the TDM Translation Table (TXT) indexed by TxCi is fetched which contains the following fields:

TxMode contains an indicator of encapsulating mode according to the following values:

0 = FRF5 encapsulation
1 = FRF8 transparent mode
2 = FRF8 translation mode

TxBecn indicates whether the last outgoing frame on the Ci had its BECN field set.
TxHdr contains 4 bytes of Frame Relay header.
TxDe indicates whether ATM CLP information goes into the FRF5 Frame Relay DE field.
TxClp indicates whether the ATM CLP is set from the FRF5 Frame Relay DE field or to 0 or to 1.

If TxMode is FRF5 encapsulation, the AAL-5 PDU is reassembled into a Frame Relay PDU, the DE field is set according to TxDe and the FECN bit is set according to TxCiBecn and the frame is sent to the Frame Interface.

As the cells are taken from the CiPtr queue, TdmReas calls the RlsBf routine in the CellPtrs block to return their BN's back to the FreeQ.

When the Frame is completely sent to the Frame Interface, TrmReas again checks the TrmOut FIFO. If the FIFO is not empty, another frame is extracted from its CiPtr, processed and sent to the TDM Interface. Note that this process allows TrmReas to completely process one Frame at a time so it is not required to interleave between frames.

6. Data Structures

The operations described in the previous section perform operations on a set of data tables. These tables are summarized below.

Port Numbers	Port Type
0	Scheduler
1 - 64	Utopia PHY
65 - 1022	TDM
1023	Local Port

Client Number	Client Type
0	CPU
1	ATM
2	TDM
3,4	Scheduler

Description of Time Slot Table (TST)

The Time Slot Table (TST) is indexed by channel (port) number. There are a total of 512 channels, organized as 16 ports times 32 channels. Each entry in the table is composed of the following fields:

Status (3-bits)
- 0=idle
- 1=sync
- 2=1st byte header
- 3=2nd byte header
- 4 =3rd byte header
- 5=4th byte header
- 6=payload
- 7=bonding base

Ci (or header bytes)
Offset
Frame checksum
AAL5 checksum
Byte fragment

Tables sorted by key

Table	Register	Width	Estimated Size
URT	UrtBase	44 b	10 KB
TRT	TrtBase	42 b	10 KB

Tables indexed by Ci

Table	Register	Width	Estimated Size
CiPtr	CiPtrBase	64 b	10 KB
UXT	UxtBase	28 b	6 KB
TXT	TxtBase	44 b	12 KB
ST	StBase	51 b	10 KB
XT	XtBase	24b	6 KB

Tables indexed by Channel (Port) Number

TST	TstBase	76 b	5 KB

Tables indexed by BN

Table	Register	Width	Estimated Size
BPtr	BptrBase	12	6 KB
BPld	BpldBase	52	156 KB

FIFO

The FIFO are stored in off-chip RAM. The FIFO sizes will be decided after simulation.

FIFO	Width	Estimated Size
TrmOut	16b	TBD
PktOut	16b	TBD
CellOut	16b	TBD
LocalOut	16b	TBD

7. Interface Signals

Mako will initially be packaged in an FPGA with attached SDRAM. The following table defines interface signals on the FPGA.

Pin(s)	I/O	Signal Name	Description
	I	Clk	Mako clock
	I	Rst_n	Mako reset (0=reset)
	I	RegRwEnb	Register access enable; 1=enable, 0=disable
	I	RegAddr5-0	Register address (bits 5-0)
	I	RegRwMode	Register read/write mode; 1=read, 0=write
	I	ArmWait_n	Register read wait; 1=inhibit read, 0=read
	I/O	RegData	Register data (bits 31-0)
	O	DbgReg	Debug register (bits 31-0)
	I	IntRq	Interrupt request
	I	RA19-RA0	SRAM address (bits 19-0)
	I/O	RD31-RD0	SRAM data (bits 31-0)

RE0-RE3	??????
T1TxClk	TDM interface 1 transmit clock
T1TxSync	TDM interface 1 transmit sync
T1TxDat	TDM interface 1 transmit data
T1RxClk	TDM interface 1 receive clock
T1RxSync	TDM interface 1 receive sync
T1TxDat	TDM interface 1 receive data
T2TxClk	TDM interface 2 transmit clock
T2TxSync	TDM interface 2 transmit sync
T2TxDat	TDM interface 2 transmit data
T2RxClk	TDM interface 2 receive clock
T2RxSync	TDM interface 2 receive sync
T2TxDat	TDM interface 2 receive data
U1TxClk	Utopia-2 interface 1 transmit clock
U1TxClAv	Utopia-2 interface 1 transmit cell available
U1TxDat0-7	Utopia-2 interface 1 transmit data (7-0)
U1TxEnb	Utopia-2 interface 1 transmit enable
U1TxPhy0-3	
U1TxSoc	Utopia-2 interface 1 transmit start-of-cell
U1RxClk	Utopia-2 interface 1 receive clock
U1RxClAv	Utopia-2 interface 1 receive cell available
U1RxDat0-7	Utopia-2 interface 1 receive data (7-0)
U1RxEnb	Utopia-2 interface 1 receive enable
U1RxPhy0-3	
U1RxSoc	Utopia-2 interface 1 receive start-of-cell
U2TxClk	Utopia-2 interface 2 transmit clock

U2TxClAv	Utopia-2 interface 2 transmit cell available
U2TxDat0-7	Utopia-2 interface 2 transmit data (7-0)
U2TxEnb	Utopia-2 interface 2 transmit enable
U2TxPhy0-4	
U2TxSoc	Utopia-2 interface 2 transmit start-of-cell
U2RxClk	Utopia-2 interface 2 receive clock
U2RxClAv	Utopia-2 interface 2 receive cell available
U2RxDat0-7	Utopia-2 interface 2 receive data (7-0)
U2RxEnb	Utopia-2 interface 2 receive enable
U2RxPhy0-4	Utopia-2 interface 2 phy 0-4 select
U2RxSoc	Utopia-2 interface 2 receive start-of-cell

8. Interface Registers

Registers in Mako vary in size from 8 bits to 32 bits. Outside of Mako, registers appear as 32-bit values. To the ARM processor, Mako registers are memory-mapped. Access to the registers is via the following three items:

MakoRWMode specifies if a register is to be read or written. A '1' denotes a read operation.

MakoRegEnab specifies that a register is to be accessed. A '1' is an enabling condition.

The Command Register (0x00) contains the register number to be accessed.

ArmWait_n, when low causes Mako to ignore read/write requests.

Mako contains the following interface registers:

Offset	Function	Name	Description
0x00	Read	Version	Version number
0x00	Write	Command	Command Register
0x01	Read/Write	RamAdr	Auto-incrementing RAM address pointer
0x02	Read/Write	RamDat	RAM data
0x03	Read/Write	BptrBase	Address of first BPtr

0x04	Read/Write	BpldBase	Address of first BPld
0x05	Read/Write	CiPtrBase	Address of first CiPtr
0x06	Read/Write	XtBase	Address of first XT entry
0x07	Read/Write	StBase	Address of first ST entry
0x08	Read/Write	BtBase	Address of first BT entry
0x09	Read/Write	UrtBase	Address of first URT entry
0x0A	Read/Write	CellOutBase	Address of first CellOut FIFO entry
0x0B	Read/Write	UxtBase	Address of first UXT entry
0x0C	Read/Write	UrtBase	Address of first FRT entry
0x0D	Read/Write	TdmOutBase	Address of first TdmlOut FIFO entry
0x0E	Read/Write	TxtBase	Address of first FXT entry
0x0F	Read/Write	SPBase	Address of first Scheduler Register Pointer block
0x10	Read/Write	TstBase	Address of first Frame Channel entry
0x11	Read/Write	NumUrt	Number of URT entries
0x12	Read/Write	NumFrt	Number of FRT entries
0x13	Read/Write	FreeQFrst	First free BN
0x14	Read/Write	FreeQLast	Last free BN
0x15	Read/Write	KeyMs	Most significant 32 bits of Search Key&Ci
0x16	Read/Write	KeyLs	Least significant 32 bits of Search Key&Ci
0x17	Read/Write	IntStat	Interrupt status
0x18	Read/Write	IntMsk	Interrupt enable mask
0x19	Read/Write	UnkFrm	Counter of Frames with unrecognized DLCI & port
0x1A	Read/Write	BadFrm	Counter of Frames with CRC errors
0x1B	Read/Write	OvfFrm	Counter of Frames lost due to buffer overflow
0x1C	Read/Write	UnkCell	Counter of Cells with unrecognized DLCI & port
0x1D	Read/Write	BadCell	Counter of Cells with CRC errors

0x1E	Read/Write	OvfCell	Counter of Cells lost due to buffer overflow
0x1F			(unassigned)
0x20	Read/Write	UtCfg	Utopia Configuration
0x21	Read/Write	PqtBase	
0x22	Read/Write	PbtBase	
0x23	Read/Write	TDMBitErr	
0x24	Read/Write	Led	
0x25	Read/Write	PtFqIn	Port Free Queue
0x26	Read/Write	PtFqOut	Port Free Queue
0x27	Read/Write	CiFqIn	Ci Free Queue
0x28	Read/Write	CiFqOut	Ci Free Queue
	Read/Write		

8.1. Version Register (0x00)

The Mako Version register is an 8-bit read-only register. It reflects the design version of Mako. The initial version number will be a binary '00000001' and will be incremented by one for each subsequent revision.

8.2. Command Register (0x00)

The Command Register is a 32-bit write-only register which specifies which one of the Mako registers is to be read/written. Setting bit 0 specifies that register 0 is to be accessed. The MakoRegEnab signal, if high, specifies that the register specified by the Command Register can be accessed. The MakoRWMode signal, if high, specifies that the register is to be read. The bits in the Command Register are mutually exclusive. Only one bit should be set. Unpredictable results may occur if more than one bit is set to a one at the same time.

8.3. RamAdr Register (0x01)

RamAdr is a 32-bit memory address in the external SDRAM.

8.4. RamDat Register (0x02)

RamDat is a 32-bit register with data to be written to the x32 SDRAM

8.5. BPtrBase (0x03)

BptrBase is a 32-bit address of the BPtr table in SDRAM.

8.6. BPldBase (0x04)

BpldBase is the 32-bit address of the start of the BPld table in SDRAM.

8.7. CiPtrBase (0x05)

CiPtrBase is the 32-bit address of the start of the CiPtr table in SDRAM.

8.8. XtBase (0x06)

XtBase is the 32-bit address of the start of the Xt table in SDRAM.

8.9. StBase (0x07)

StBase is the 32-bit address of the start of the St table in SDRAM.

8.10. BtBase (0x08)

BtBase is the 32-bit address of the start of the Bt table in SDRAM.

8.11. UrtBase (0x09)

UrtBase is the 32-bit address of the start of the URT table.

8.12. CellOutBase (0x0A)

8.13. UxtBase (0x0B)

8.14. TrtBase (0x0C)

8.15. TdmOutBase (0x0D)

8.16. TxtBase (0x0E)

8.17. SPBase (0x0F)

8.18. TstBase (0x10)

8.19. NumUrt (0x11)

Number of entries in the URT table. This is a 16-bit quantity.

8.20. NumFrt (0x12)

Number of entries in the FRT table. This is a 16-bit quantity.

8.21. FreeQFrst (0x13)

FreeQFrst is the entry number of the first free buffer in BN. This is a 16-bit quantity.

8.22. FreeQLast (0x14)

FreeQLast is the entry number of the last free buffer in BN. This is a 16-bit quantity.

8.23. KeyMs (0x15)

8.24. KeyLs (0x16)

8.25. IntStat Register

The IntStat Register contains event flip-flops which are set on occurrence of significant system events. InStat is an 8-bit register with the bits defined in the table below.

Bit	Description
0	Unknown frame count >= 128
1	Bad frame count >= 128
2	Overflow frame count >= 128
3	Unknown cell count >= 128
4	Bad cell count >= 128
5	Overflow cell count >= 128

8.26. IntMsk Register

The IntMsk Register bits correspond with the interrupt event bits in the IntStat register and control the ability of specific events to generate a Mako interrupt to the Host CPU. Each bit in the IntMsk Register is logically and'ed with the corresponding bit in the IntStat Register. If the result is non-zero then the interrupt request will be passed to the Host.

8.27. UnkFrm Register

The UnkFrm register is an 8 bit counter of the number of frames discarded because their DLCI and Port didn't correspond to any defined VC. It sticks at all 1's. It clears to all 0's after being read through the CPU interface. A maskable interrupt request bit is set when the count reaches 128. This gives the software time to react and read the count before the error count reaches 255. If the count reaches 255, it will remain at 255 until read by the CPU regardless of the number of additional error events.

8.28. BadFrm Register (0x1A)

The BadFrm register is an 8 bit counter of the number of frames discarded because CRC was incorrect. It

sticks at all 1's. It clears to all 0's after being read through the CPU interface.

8.29. OvfFrm Register (0x1B)

The OvfFrm register is an 8 bit counter of the number of frames discarded because there were no buffers to hold them. It sticks at all 1's. It clears to all 0's after being read through the CPU interface.

8.30. UnkCell Register (0x1C)

The UnkCell register is an 8 bit counter of the number of cells discarded because their VPI, VCI and the most significant bit of their PTI didn't correspond to any defined VC. It sticks at all 1's. It clears to all 0's after being read through the CPU interface.

8.31. BadCell Register (0x1D)

The BadCell register is an 8 bit counter of the number of cells discarded because HEC was incorrect. It sticks at all 1's. It clears to all 0's after being read through the CPU interface.

8.32. OvfCell Register (0x1E)

The OvfCell register is an 8 bit counter of the number of cells discarded because there were no buffers to hold them. It sticks at all 1's. It clears to all 0's after being read through the CPU interface.

8.33. UtCfg Register (0x1F)

The UtCfg register is a 32 bit register that controls the configuration of the Utopia ports. The functions of the various fields within UtCfg are described in the following table.

Bit(s)	Description
0 - 4	Utopia PHY number. This field is ignored if the Utopia is a master or is Utopia level 1. In Utopia level 2 slave mode, the port will respond when this address is polled by the master.
5	Utopia level: 0 for level 1, 1 for level 2.
6	Utopia Master/Slave: 0 for Slave, 1 for Master.
7	Utopia Bus Width: 0 for 8 bit, 1 for 16 bit. Note: current Mako only supports 8 bit.
8 - 12	Utopia PHY number. This field is ignored if the Utopia is a master or is Utopia level 1. In Utopia level 2 slave mode, the port will respond when this address is polled by the master.
13	Utopia level: 0 for level 1, 1 for level 2.
141	Utopia Master/Slave: 0 for Slave, 1 for Master.
5	Utopia Bus Width: 0 for 8 bit, 1 for 16 bit. Note: current Mako only supports 8 bit.
16 - 20	Utopia PHY number. This field is ignored if the Utopia is a master or is Utopia level 1. In Utopia level 2 slave mode, the port will respond when this address is polled by the master.

21 Utopia level: 0 for level 1, 1 for level 2.

22 Utopia Master/Slave: 0 for Slave, 1 for Master.

23 Utopia Bus Width: 0 for 8 bit, 1 for 16 bit. Note: current Mako only supports 8 bit.

24 - 28 Utopia PHY number. This field is ignored if the Utopia is a master or is Utopia level 1. In Utopia level 2 slave mode, the port will respond when this address is polled by the master.

29 Utopia level: 0 for level 1, 1 for level 2.

30 Utopia Master/Slave: 0 for Slave, 1 for Master.

31 Utopia Bus Width: 0 for 8 bit, 1 for 16 bit. Note: current Mako only supports 8 bit.

9. Functional Block Descriptions

The following components and sub-systems provide the functionality to support the test functions specified above.

9.1. CellPtrs

The CellPtrs block is actually packaged inside the PointerSwitch block. It consists of threaded lists of BN's and routines to move them from one queue to another. Empty BN's are kept in the FreeQ. BN's containing Cells are linked to per-Ci queues. The following figure is a generic block diagram applying to the routines in the CellPtrs block.

As implied in the figure, there are multiple request, acknowledge, ClId and Ci inputs. There is one set per client. A calling client places its ClId and the Ci to be affected onto the ClId and Ci busses to which it is connected. It requests the routine's service by asserting its BfRq. When the routine has serviced the client, it asserts BfAk. Note that BfRq and BfAk are generic. The actual signals will be GetBfRq, RlsBfRq, etc. depending on which routine is being called. Outputs from the CellPtrs block consist of address, data, request and acknowledge signals to access SRAM.

The GetBf routine extracts one BN from the FreeQ and places it on the Ci queue.

The RlsBf routine extracts one BN from the Ci queue and puts it back on the FreeQ.

All of the CellPtr routines use a pair of pointer registers and two sets of SRAM based pointers. FreeQFrst and FreeQLast are registers which indicate the first and last BN's in the free queue. Upon initialization, these are set to the lowest and the highest numbered BN, respectively.

9.2. TDM Interface

This interfaces to TDM highway that supports up to 4 T1/E1 line interface units (LIU).

9.3. TdmRes

The TdmRes block diagram is shown in the following figure:

FrmDatAvail, FrmSof, FrmDatRdClk and FrmDatIn are signals associated with the Frame I/F. FrmSegRq, FrmSegAk, FrmDatOut and Ci are interface signals to the TdmSeg block. The signals on the right side provide access to SRAM.

9.4. TdmSeg

The TdmSeg block segments frame payloads into streams of AAL-5 encoded ATM cells with zero fill, as necessary, and AAL-5 trailer. While processing the frame, TdmSeg verifies the frame CRC. If the CRC is bad, the frame is discarded and the BadFrm counter is incremented.

Each cell is assigned a BN via a call to GetBf. If no BN is available, the frame is discarded and the OvfFrm counter is incremented. If a BN is available, GetBf adds a BPtr to the Ci's queue, leaving the PTI and ClldSel fields zero. TdmSeg fills in the PTI field. TdmSeg then stores the cell payload data in the associated BPld entry.

9.5. TdmReas

The TdmReas (Frame Reassembly) block receives BfNm's from the PointerSwitch block. When TdmReas receives a BfNm, it examines the associated cell to see whether it is the last cell of a frame payload. If not, it simply leaves the new BfNm on its queue. If it is the last cell of a frame PDU, TdmReas extracts all the cells of the PDU from its queue and forms the outgoing frame, including encapsulation headers. In the process, it releases the BfNm's of all the associated cells back to the FreeQ.

9.6. Cell Interface

The Cell Interface block is a Utopia interface, selectable to level 1 or level 2 by the UtCfg register.

9.7. CellRes

The CellHdrRes block examines the VPI, VCI and PTI fields of incoming cells and associates them with Ci's.

9.8. CellXlt

9.9. PointerSwitch

9.10. CellSched

10. Software Interface

The ARM microprocessor interfaces with Mako over a Local Bus. The Local Bus interface runs at 33 MHz and connects the ARM CPU to various peripherals, including the Mako. The Local Bus interface is used to pass management and control information to and from the ARM to the various communication ports as well as providing configuration, control and status registers and access to the Mako RAM tables. The registers are memory mapped to the ARM memory address space.

PAGE 142 INTENTIONALLY BLANK

```
-- Sch.vhd
--
-- This is the module that applies quality of service to cell streams.
-- It is part of Mako versions 0x30000002 and above.
--
-- There may be many ports with widely different line rates. In order
-- to assure that each port is serviced with a relative frequency
-- commensurate with its relative rate, the host CPU builds a port
-- service sequence table, PtSq, in DRAM. It is a sequence of port
-- numbers, with port numbers of fast ports occuring more frequently
-- than port numbers of slow ports. The Pq state machine reads the
-- PtSq and queues up a service list in the PtSq FIFO.
--
-- Incoming QSw requests in the Sqt are qualified by the Cq state machine.
-- For those that qualify, i.e. whose Ci's are not already active or are
-- CT/QFC blocked, their Ci's are put into a CiAq FIFO for activation.
--
-- Ci Activation and cell queing and are performed by the Sch state machine.
-- It alternates between activating Ci's and queing cells.
--
-- Cell queing is actually cell sequencing. The scheduler determines
-- the sequence in which a mix of data and idle cells is sent.
-- Scheduling is performed by the output port logic at the port line
-- rate. Each Ci is given priority and allocated a ratio of of transmit
-- opportunities based on its assigned QoS and programmed rate.
--
-- The core element of the scheduler is the Bubble Table that contains
-- timed and prioritized set of cell transmit requests for one port at
-- a time. It is actually a group of sub tables. Each sub table contains
-- requests for one QoS within the port. The first priority decision
-- is between QoS's and the second is between Ci's based on next target
-- time and programmed rate.
--
-- The highest priority cell is queued for sending by placing it's Bn
-- in CiPtr(Ci)->Sch. The sequence of queued cells begins with the
-- Bn in CiPtr(Ci)->Out and chains through BPtr(Bn) until reaching
-- the last scheduled cell's Bn in CiPtr(Ci)->Sch. After a cell is
-- queued up, the next target time is calculated by adding the interval
-- time for the Ci to the previous target time. Target time and
-- Interval time are formatted as fixed point fractions, thus allowing
-- effective line rates to be specified with fine granularity.
--
-- After the next target time is calculated, a binary search is made
-- within the QoS sub table based on target time and Ci rate. If
-- multiple Ci's in the same QoS have the same target time, they are
-- prioritized by rate; faster rate results in higher priority.
```

--
-- After the comparison search, if the Ci entry is no longer the highest
-- priority, it will be removed from the Bubble Table by shifting all
-- Bubble Table entries above it down by one. If the Ci is to remain
-- enabled, a new place for the Ci entry will be created by shifting
-- the entry in the new place and all entries above it up by one place.
-- The Bubble Table architecture is such that this shifting process
-- is very fast. A small number of clock cycles (as few as one) to
-- prepare and one clock cycle to execute the shift.
--
-- Ci's are activated or deactivated dynamically based on cell
-- availibility and QoS criteria. If there are no cells in the CiPtr(Ci)
-- queue, the Ci will be disabled. If there are cells in the queue
-- criteria to enable the Ci depend on QoS in priority order as follows:
--
-- Pacing CBR:
-- Ci enabled anytime any data Ci is enabled on the port. These virtual
-- streams generate idle cells on the port to throttle data when the
-- port has an artificially reduced line speed. Multiple Pacing CBR Ci's
-- are allowed to achieve fine granularity in reduced line rate.
--
-- CBR:
-- Ci always enabled. This QoS will not be overbooked by software.
--
-- VBRrt or VBRnrt:
-- Ci Enabled unless it has filled its limit for cells within a measure.
-- A measure is a count of transmit opportunities representing about
-- 125 msec. This count will vary according to the line rate of the
-- of the port. The SCR of this QoS will not be overbooked.
--
-- VBRrt or VBRnrt:
-- Ci Enabled unless it has filled its limit for cells within a measure.
--
-- ABR
-- Ci is always enabled. ABR is like CBR except that it is lower
-- priority and the effective PCR can change dynamically.
--
-- QFC/CT
-- Ci is enabled until the Tx credit limit is reached. Once disabled
-- because of credit limit, it can be reenabled by an incoming credit
-- update cell.
--
-- UBR
-- Ci always enabled. UBR is treated like CBR except that it is the
-- lowest priority. UBR is actually UBR++ since it always has a PCR.
-- It is equivalent to UBR if the PCR is programmed at line rate.

```
--
-- ThVBRrt
--   Ci always enabled. VBRrt Ci's are moved to this QoS when their limit
--   of cells per measure is reached.
--
-- ThVBRnrt
--   Ci always enabled. VBRnrt Ci's are moved to this QoS when their limit
--   of cells per measure is reached.
--
-- Several tables and registers are used by the scheduler. Some are stored
-- in DRAM, others in SRAM.
--
-- There are three imbedded SRAM blocks.
--
--  PtSq block: 8 bits wide x 256 deep. Each byte is seen by the host
--           CPU as the significant 8 bits of a Dword. These dwords
--           occupy 0x400000 - 0x4000FF in Mako Ram address space.
--
--  SchPt block: 106 bits wide x 64 deep. Only the least significant
--           26 bits of each entry is visible in Mako Ram address
--           space. These are at addresses 0x400100 - 0x40013F.
--
--  SchBt block: 1440 wide x 64 deep. These are not visible in Mako Ram
--           address space.
--
-- The following tables are used:
--
-- Switch Request Related:
--
--  SqT (Switch request Queue Table)
--
--    SqT contains one entry per port.
--
--    Each entry in SqT contains the following fields:
--
--      SqT(SwClt)->In  (wd 0 bits  0 - 15) = number of newest Sr
--      SqT(SwClt)->Out (wd 0 bits 16 - 31) = number of oldest Sr
--      SqT(SwClt)->In  (wd 1 bits  0 - 15) = number of Sr's pending for port
--
--
--  SrT (Switch Request Table) in DRAM
--
--    SrT contains one entry per Switch Request Buffer
--
--    Each SrT entry contains the following fields:
--
```


14.

```
--      SrT(Sm)->Nxt (bits  0 - 15) = Next Switch Request# in chain
--      SrT(Sm)->Ci  (bits 16 - 24) = Ci of the cell being switched
--      SrT(Sm)->unused (bits 25 - 31) = unused
--
-- Cell Buffer related
--
--  CiPtr (Ci Pointers) in DRAM
--
--    CiPtr points to cells threaded on per Ci queues.
--
--    Each CiPtr entry contains the following fields:
--
--      CiPtr(Ci)->In  (bits  0 - 15 of 1st word) = Bn of newest cell
--      CiPtr(Ci)->Out (bits 16 - 31 of 1st word) = Bn of oldest cell
--      CiPtr(Ci)->Ocp (bits  0 - 15 of 2nd word) = Bn's occupied by this Ci
--      CiPtr(Ci)->Sch (bits 16 - 31 of 2nd word) = Last Bn processed by Sched
--
--
--  Bptr (Buffer Pointers) in DRAM
--
--    BPtr contains chaining pointers for Bn's that are on the various
--    buffer queues.
--
--    Each BPtr entry contains the following field:
--
--      BPtr(Bn)->Nxt    (bits  0 - 15) = Next Bn in chain
--      BPtr(Bn)->unused (bits 16 - 31) = unused
--
--
--  BPld (Buffer Payloads) in DRAM
--
--    BPld contains the headers and payloads of the cells on the various queues
--
--    Each BPld entry contains the following field:
--
--      BPld(Bn)->Hdr (1st word)        = Cell Header
--      BPld(Bn)->Pld (2nd - 13th word) = Cell payload
--
-- Scheduler Port Sequencing related:
--
--  PtSq (Port Sequence Table) in Sram
--
--    Software calculates the sequence in which ports need to be serviced and
--    builds the sequenced list of port buffer numbers into the PtSq. Entries
--    are 1 byte each. The first byte in the table contains the total number
--    of entries, which begin with the second byte of the table.
```

```
--
-- PtSqWd (Port Sequence Word Number) register
--
--   PtSqWd contains the current PtSq word number. When the value is 0
--   the first word of PtSqWd is fetched, refreshing knowledge of the
--   number of entries in the table since lower 10 bits of the first word
--   contain the total number of entries in the table.
--
-- PtSqOfst
--
--   PtSqOfst is the offset of the current port sequence number within the
--   currently cached 31 bit word. 0 = first (need to read), 1 = 2nd, 2 = 3rd.
--
-- PtSqMax
--
--   PtSqMax holds the total number of port numbers in the PtSq.
--
-- Active Port Related:
--
-- PtSq (Port Scheduler Queue) fifo
--
--   This is a fifo of port numbers that are active and need to be scheduled.
--
-- SchPtBf
--
--   This register contains the current port number buffer associated with
--   the scheduler.
--
-- SchPt, SchPtRg (Scheduler per-Port Table) in SRAM and registers
--
--   The SchPt Table contains an entry per Port. Up to 64 entries are
--   in SRAM. The active entry is in SchPtRg which corresponds to SRAM
--   buffer number PtBfNum. SchPt(PtBfNum)->Pt tells the Pt. Host
--   software allocates the PtBfNum's to the ports and initializes it by
--   zeroing the entire table, then writing port numbers into the
--   SchPt(PtBfNum)->Pt fields. It is stored as one wide parallel
--   word in SRAM but is accessible to the host CPU through the DRAM
--   registers as though it were stored as contiguous little endian
--   32 bit words.
--
--   SchPt(PtBfNum)->PcbrCnt   (bits 105 downto 104) = Number of Pacing Constant
Bit Rate Ci's (max=4)
--   SchPt(PtBfNum)->CbrCnt    (bits 103 downto  98) = Number of Active Constant
Bit Rate Ci's
--   SchPt(PtBfNum)->VbrrtCnt  (bits  97 downto  92) = Number of Active Variable
Bit Rate real time Ci's
```

```
--   SchPt(PtBfNum)->VbrnrtCnt   (bits 91 downto 86) = Number of Active Variable
Bit Rate not real time Ci's
--   SchPt(PtBfNum)->AbrCnt      (bits 85 downto 80) = Number of Active Available
Bit Rate Ci's
--   SchPt(PtBfNum)->QfcCnt      (bits 79 downto 74) = Number of Active QFC Ci's
--   SchPt(PtBfNum)->UbrCnt      (bits 73 downto 68) = Number of Active UBR Ci's
--   SchPt(PtBfNum)->UbPlsrCnt   (bits 67 downto 62) = Number of Active UBR Ci's
--   SchPt(PtBfNum)->ThVbrrtCnt  (bits 61 downto 56) = Number of Active Throttled
VBRrt Ci's
--   SchPt(PtBfNum)->ThVbrnrtCnt (bits 55 downto 50) = Number of Active Throttled
VBRnrt Ci's
--   SchPt(PtBfNum)->ThUbrPlsCnt (bits 49 downto 44) = Number of Active Throttled
VBRnrt Ci's
--   SchPt(PtBfNum)->MsrCnt      (bits 43 downto 28) = Current PIC counts in the
measure
--   SchPt(PtBfNum)->Pic         (bits 27 downto 14) = Port Interval Counter
--   SchPt(PtBfNum)->MsrPwr2     (bits 13 downto 10) = Log2 of max counts in a
measure
--   SchPt(PtBfNum)->Pt          (bit  9 downto  0) = Port number
--
-- Active Ci Related:
--
--  CiAq (Ci Activation Queue) fifo
--
--   This is a fifo of Ci numbers that need to be activated.
--
--  CiStat, CiStatRg (Ci Status) in Dram, 16 Ci's per 32 bit word.
--  This table is located at offset 0x8000 above the start of SchCi.
--
--    CiStat(Ci)->Act      (bit 0) = Ci active
--    CiStat(Ci)->QfcBlkd  (bit 1) = Ci is QFC & out of Xmt credits
--
--  CiStatNum (Ci Status Number) register
--
--    Bits 4 and up of Ci numbers cached in CiStatRg
--
--  SchCi (Scheduler per-Ci Table) in Dram, cached in a register.
--  This table is located at the address pointed to by PI_SchBase.
--
--    Each entry consists of 4 dwords.There is an entry for each Ci.
--    Ci 0 is reserved. Ci 1 through 8191 are the available assignable
--    Ci's. Ci's above 8192 are reserved for PCBR Ci's. There are 4
--    such entries for each port. In the PCBR Bt entry the Ci number
--    shall be in the range 0 - 3.  The overall Ci number is equal to
--    8191 + 4*Port Number + Bt Ci number.  Thus Ci numbers for PCBR
--    are pre-allocated.
```

```
--
--  The first dword has the same function for all QoS's.
--  Subsequent words depend on QoS with the exception that bit 31
--  of the 4th word is the trottled status bit. The 1st word is
--  only written by the host PC. The 4th word is only written by
--  Mako. The writer of words 2 and 3 depends on the QoS.
--
--  SchCi(Ci)->ThrtId  (Wd 0 bit 31)          = When set it means that this Ci is throttled
--  SchCi(Ci)->Unused  (Wd 0 bits 30 downto 28)
--  SchCi(Ci)->FrctTm  (Wd 2 bits 27 downto 25) = Fractional portion of TgtTm
--  SchCi(Ci)->CiPtBf  (Wd 0 bits 24 downto 18) = SRAM buffer number of the
destination port for the Ci
--  SchCi(Ci)->ItInt   (Wd 0 bits 17 downto  6) = 12 bit integer portion of Interval Time
--  SchCi(Ci)->ItFrct  (Wd 0 bits  5 downto  3) = 3 bit fractional portion or Interval
Time
--  SchCi(Ci)->QoS     (Wd 0 bits  2 downto  0) = Quality of Service for Ci, 0=PCBR,
1=CBR, 2=VBRrt, 3=VBRnrt,
--                                      4=ABR, 5=QFC, 6=UBR+ & 7=UBR
--
--  for PCBR, CBR and UBR
--  SchCi(Ci)->Unused  (Wd 1 bits 31 downto  0)
--  SchCi(Ci)->Unused  (Wd 2 bits 31 downto  0)
--  SchCi(Ci)->Unused  (Wd 3 bits 31 downto  0)
--
--  for VBRrt and VBRnrt
--  SchCi(Ci)->MsrCnt  (Wd 1 bits 31 downto 16) = The number of cells sent during
this Time Measure Unit
--  SchCi(Ci)->MsrMax  (Wd 1 bits 15 downto  0) = Max cells to be sent in a Time
Measure before throttling
--  SchCi(Ci)->Unused  (Wd 3 bits 31 downto 16)
--  SchCi(Ci)->MsrRoll (Wd 2 bits 15 downto  0) = Rollover count of Time Measure
Unit that sent count represents
--
--  for ABR
--  SchCi(Ci) Wd 1,2,3 TBD
--
--  for QFC/CT
--  SchCi(Pt)->RxCnt   (Wd 1 bits 31 downto 24) = Rx count
--  SchCi(Pt)->TxCnt   (Wd 1 bits 23 downto 16) = Tx count
--  SchCi(Pt)->RxQtm   (Wd 1 bits 15 downto  8) = Rx Quantum
--  SchCi(Pt)->TxLmt   (Wd 1 bits  7 downto  0) = Tx limit
--  SchCi(Ci)->Unused  (Wd 2 bits 31 downto  0)
--  SchCi(Ci)->Unused  (Wd 3 bits 31 downto  0)
--  Note: SchCi(0) contains QFC info for the port
--
--  SchCiNum (Ci Number register)
```

--
-- This internal register has the number of the Ci whose SchCi entry is loaded in the SchCi register.
--
-- SchCiNum->(bits 13 downto 0)= The Ci represented by the loaded SchCi
--
-- Cell Scheduling Time Related:
--
-- Bt (Bubble Table) in SRAM and registers
--
-- Each SchBt entry is a request to queue a cell for transmit on a Ci
-- at a target time. There is an ordered group of entries per QoS.
-- There is an ordered set of QoS groups per port. In other words,
-- the SchBt contains three hierarchies of ordering: by port; by Qos;
-- by cell target time.
--
-- The SchBt is physically partitioned into pages of 32 entries each.
-- 64 pages are cached in SRAM. The active page is cached and
-- processed in registers.
--
-- Each SchBt entry contains the following fields:
--
-- SchBt->TtInt (bits 44 downto 31) = Integer portion of Target Time
-- SchBt->TtFrct (bits 30 downto 28) = Fractional portion of Target Time
-- SchBt->IvInt (bits 27 downto 16) = Integer portion of Interval Time
-- SchBt->IvFrct (bits 15 downto 13) = Fractional portion of Interval Time
-- SchBt->Ci (bits 12 downto 0) = Ci
--
-- The SchBt is initialized to all zero. Segments of the page in
-- registers can be shifted up and down for insertion and deletion
-- of entries. The most significant bit of all the SchBt->TfInt fields
-- in the register can be concurrently zeroed in one clock cycle.
--
-- PUC (Port Useage Count) 64 Sram entries of 16 bits
--
-- These counters contain the count of MIC when this port last sent a cell. When the MIC overflows,
-- the MIC is reset to 0x0100 and the upper 8 bits of each PUC is shifted to the lower 8, with the upper
-- being set to 0.
--
-- PUC(entry)->Mic (bits 15 downto 0) =MIC value when last cell sent for Port
--
-- FullMic (Full Mastet Interval Counter)
-- (64 bit register, 2 double words)
--

```
--    This complete counter will record the number of cells transmitted, less 256 for every
65,536
--    because when the lower 16 bits overflow they are reset to 0x0100 instead of zero.
This is
--    to retain proper ordering on the PUC's.
--
--    FullMic->MicOfl (bits 63 downto 16) = Overflow count of MIC
--    FullMic->Mic    (bits 15 downto 0)  = Value placed in PUC
--
--  This 10 bit register keeps count of the number of active ports at the moment.
--
-- NumActPt->(bits 9 downto 0)
--
-- CqSt qualifies cells for scheduling and puts qualified cells in Ci
-- activation queue.
--
-- CqSt00
--   if((CiAqFull) or (not SwAlert and not (CiAqFullDlyd and not CiAqFull)))
--     goto CqSt00
-- CqSt01
--   if(CqCi <= Sqt(0)->Out = 0) -- Using CqCi to hold Sr temporarily
--     goto CqSt00
-- CqSt02
--   RlsSr(0)
-- CqSt03
--   if(CqCi <= Srt(CqCi)->Ci = 0)
--     goto CqSt01
-- CqSt04
--   if(CqCi bits 4 and above = CiStatNum)
--     goto CqSt06
--   assert RamMore
--   CiStat(CiStatNum) <= CiStatRg
--   CiStatNum <= CqCi>>4
-- CqSt05
--    CiStatRg <= CiStat(CiStatNum)
-- CqSt06
--   if(CiStatRg(CqCi(3:0))->Act /= 0 or CiStat(CqCi(3:0))->Qfc /= 0)
--     goto CqSt00
--   add CqCi to CiAq fifo
--   CiStat(CqCi)->Act <= 1
--   goto CqSt00
--
-- SchSt activiates Ci's into the Bubble Table and schedules Ci's on a per
-- port basis.
--
-- Static
```


15

```
-- Offsets in BTbl of the Qos groups
--   BtPcbr    <= 0
--   BtCbr     <= BtPcbr   + SchPtRg->PcbrCnt
--   BtVbrrt   <= BtCbr    + SchPtRg->CbrCnt
--   BtVbrnrt  <= BtVbrrt  + SchPtRg->VbrrtCnt
--   BtAbr     <= BtVbrnrt + SchPtRg->VbrnrtCnt
--   BtQfc     <= BtAbr    + SchPtRg->AbrCnt
--   BtUBRPls  <= BtQfc    + SchPtRg->QfcCnt
--   BtUbr     <= BtUbrPls + SchPtRg->UbrPlsCnt
--   BtThVbrrt <= BtUbr    + SchPtRg->UbrCnt
--   BtThVbrnrt<= BtThVbrrt + SchPtRg->ThVbrrtCnt
--   BtThUbrPls<= BtThVbrnrt+ SchPtRg->ThVbrnrtCnt
--  PtAct   <= BtTot /= SchPtRg->PcbrCnt
--  BtQos(QoS)<= Mux selected by QoS with BtPcbr, et al as inputs
--  if(SchCi->QoS = Pcbr)
--    BtStrt <= BtPcbr
--    BtCnt  <= SchPtRg->PcbrCnt
--  if(SchCi->QoS = Vbrrt)
--    BtStrt <= BtVbrrt
--    BtCnt  <= SchPtRg->VbrrtCnt
--  if(SchCi->QoS = Vbrnrt)
--    BtStrt <= BtVbrnrt
--    BtCnt  <= SchPtRg->VbrnrtCnt
--  if(SchCi->QoS = Abr)
--    BtStrt <= BtAbr
--    BtCnt  <= SchPtRg->AbrCnt
--  if(SchCi->QoS = fc)
--    BtStrt <= Btfc
--    BtCnt  <= SchPtRg->fcCnt
--  if(SchCi->QoS = UbrPls)
--    BtStrt <= BtUbrPls
--    BtCnt  <= SchPtRg->UbrPlsCnt
--  if(SchCi->QoS = Ubr)
--    BtStrt <= BtUbr
--    BtCnt  <= SchPtRg->UbrCnt
--  TgtTm <= SchCi->ItInt&SchCi->ItFrct + SchPt->Pic&SchCi(Ci)->FrctTm
-- SchSt00
--  PollQos <= 0
--  SchCqTm <= not CiSrvTm
--  if((SchCqTm and CiAqEmpty) or (not SchCqTm and SchPtBf = 0))
--  goto SchSt00
--  if(not SchCqTm)
--    goto SchSt0B
-- /***************\
-- * Activate a Ci *
-- \***************/
```

```
-- if(SchCiNum = 0)
--   WdCnt <= 0
--   goto SchSt02
-- if(SchCiNum = CiAq or SchCi(Ci)->QoS = (PCbr or Cbr or Abr or Ubr))
--   WdCnt <= 0
--   goto SchSt03
-- WdCnt <= 2
-- SchSt01 -- save volatile part of old SchCi entry
-- SchCiBase(4*SchCiNum + WdCnt) <= SchCiRg(WdCnt)
-- if(WdCnt /= 2)
--   WdCnt++
--   goto SchSt01
--   WdCnt <= 0
-- SchSt02 -- retrieve new SchCi entry
-- SchCiRg(WdCnt) <= SchCiBase(4*CiAq + WdCnt)
-- WdCnt++
-- if((new SchCiRg->QoS = (Vbrrt or Vbrnrt or Qfc) and WdCnt /= 1) or (new
SchCiRg->QoS = Abr and WdCnt /= 2))
--   goto SchSt02
-- SchSt03
-- SchCiNum <= CiAq
-- pop CiAq
-- if(SchPtBf = SchCiRg->CiPtBf)
--   goto SchSt07
-- SchSt04
-- SchPt(SchPtBf) <= SchPtRg -- save current SchPt
-- SchBt(SchPtBf) <= SchBtRg -- save current SchBt
-- SchPtBf <= SchCiRg->CiPtBf
-- SchSt05
-- SchPtRg <= SchPt(SchPtBf) -- retrieve new SchPt
-- SchBtRg <= SchBt(SchPtBf) -- retrieve new SchBt
-- SchSt06
-- if(TgtTm msb = 0 and SchPt->Pic msb = 1)
-- zero msb of all SchBt->TgtInt's
-- zero msb of SchPt->Pic
-- SchSt07
-- BtNdx <= SchSrch(TgtTm,BtQos(SchCi->Qos),BtCnt(SchCi->Qos))
-- BTbl(ShClr) -- clear Bt shift enables
-- SchSt08
-- BTbl(ShLdRq,BtNdx) & wait for BTbl(ShLdAk) -- set shift starting point
-- BTbl(ShUp)
-- SchSt09
-- SchBtRg(BtNdx)->TtInt&->TtFrct <= TgtTm
-- SchBtRg(BtNdx)->ItInt&->ItFrct <= SchCi->ItInt&SchCi->ItFrct
-- SchBtRg(BtNdx)->Ci <= SchCiNum
-- SchSt0A
```

```
-- if(SchCi->Qos = Cbr)
--    SchPtRg->CbrAct <= 1
--    SchPtRg->CbrCnt++
-- if(SchCi->Qos = Vbrrt)
--    SchPtRg->VbrrtAct <= 1
--    SchPtRg->VbrrtCnt++
-- if(SchCi->Qos = Vbrnrt)
--    SchPtRg->VbrnrtAct <= 1
--    SchPtRg->VbrnrtCnt++
-- if(SchCi->Qos = Abr)
--    SchPtRg->AbrAct <= 1
--    SchPtRg->AbrCnt++
-- if(SchCi->Qos = Qfc)
--    SchPtRg->QfcAct <= 1
--    SchPtRg->QfcCnt++
-- if(SchCi->Qos = Ubr)
--    SchPtRg->UbrAct <= 1
--    SchPtRg->UbrCnt++
-- goto SchSt00
-- /****************\
-- * Service a port *
-- \****************/
-- SchSt0B
--   NewPtBf <= PtSq(NxtPt)
--   NxtPt++ -- but not until after this state
--   if(PtSq(NxtPt) = SchPtBf)
--     goto SchSt0F
-- SchSt0C
--   SchPt(SchPtBf) <= SchPtRg -- save current SchPt
--   SchBt(SchPtBf) <= SchBtRg -- save current SchBt
--   SchPtBf <= NewPtBf
-- SchSt0D
--   SchPtRg <= SchPt(SchPtBf) -- retrieve new SchPt
--   SchBtRg <= SchBt(SchPtBf) -- retrieve new SchBt
-- SchSt0E
--   if(not PtAct)
--     goto SchSt00 -- done cuz no active Ci's
--   SchPtRg->Pic++
--   if(Qos = Vbrrt or Qos = Vbrnrt or Qos = UbrPls)
--     SchPtRg->MsrCnt++
--   if(bit SchPtRg->MsrPwr2 of SchPtRg->MsrCnt doesn't go from 1 to 0)
--     goto SchSt17 -- jmp if not at end of measure
--   if(SchPtRg->ThVbrrtCnt = 0 and SchPtRg->ThVbrnrtCnt = 0 and SchPtRg-
>ThUbrPlsCnt = 0)
--     goto SchSt17 -- jmp if nothing to unthrottle
--   if(SchPtRg->ThVbrrtCnt = 0)
```

```
--    goto SchSt13 -- jmp if no VBRrt to unthrottle
-- /*******************************\
-- * Unthrottle at end of measure *
-- \*******************************/
-- SchSt0F
--   BtNdx <= SchSrch(SchBtRg(BtQos(ThVbrrt))->TtInt&->IvInt&-
>IvFrct,BtQos(Vbrrt))
--   BTbl(ShClr) -- clear Bt shift enables
-- SchSt10
--   BTbl(ShLdRq,BtNdx) & wait for BTbl(ShLdAk) -- set shift starting point
--   BTbl(ShUp)
-- SchSt11
--   SchBtRg(BtNdx) <= SchBt(BtQos(ThVbrrt))
-- SchSt12
--   BTbl(ShLdRq,BtQos(ThVbrrt)) & wait for BTbl(ShLdAk) -- set shift starting point
--   BTbl(ShDn)
--   if(SchPtRg->ThVbrrtCnt /= 0)
--     goto SchSt12
--   if(SchPtRg->ThVbrnrtCnt /= 0)
--     goto SchSt17
-- SchSt13
--   BtNdx <= SchSrch(SchBtRg(BtQos(ThVbrrt))->TtInt&->IvInt&-
>IvFrct,BtQos(Vbrrt))
--   BTbl(ShClr) -- clear Bt shift enables
-- SchSt14
--   BTbl(ShLdRq,BtNdx) & wait for BTbl(ShLdAk) -- set shift starting point
--   BTbl(ShUp)
-- SchSt15
--   SchBtRg(BtNdx) <= SchBt(BtQos(ThVbrrt))
-- SchSt16
--   BTbl(ShLdRq,BtQos(ThVbrrt)) & wait for BTbl(ShLdAk) -- set shift starting point
--   BTbl(ShDn)
--   if(SchPtRg->ThVbrrtCnt /= 0)
--     goto SchSt16
-- /********************\
-- * Check for ready Ci *
-- \********************/
-- SchSt17
--   if(SchBt(BtQos(PollQos))->TtInt > SchPt->Pic)
--     if(PollQos = Ubr)
--       goto SchSt25 -- go send idle cell cuz no Ci ready
--     PollQos++
--     goto SchSt17
--   SchBtRg(BtQos(QoS))->TtInt&->TtFrct <= TgtTm
--   if(PollQos = Pcbr)
--     goto SchSt25 -- go send idle cell cuz Pcbr ready
```

```
-- /****************\
-- * Send data cell *
-- \****************/
-- SchSt18
--   assert RamMore
--   SchNxtBn <= CiPtr(SchBt(BtQos(QoS))->Ci)->Out
--   SchLstBn <= CiPtr(SchBt(BtQos(QoS))->Ci)->In
-- SchSt19
--   assert RamMore
--   SchOcp <= CiPtr(SchBt(BtQos(QoS))->Ci)->Ocp
--   if(CiPtr(SchBt(BtQoS(QoS))->Ci)->Sch = 0)
--     goto SchSt1B
-- SchSt1A
--   assert RamMore
--   SchNxtBn <= BPtr(SchBn)
-- SchSt1B
--   CiPtr(SchBt(BtQos(QoS))->Ci)->Ocp <= SchOcp
--   CiPtr(SchBt(BtQos(QoS))->Ci)->Sch <= SchNxtBn
-- SchSt1C
--   GetSr(SchBt(BtQos(QoS))->Ci)
--   if(SchNxtBn /= SchLstBn)
--     goto SchSt21
-- /***************\
-- * Deactivate Ci *
-- \***************/
-- SchSt1D
--   if(SchBt(BtQoS(QoS))->Ci bits 4 and above = CiStatNum)
--     goto SchSt1F
--   assert RamMore
--   CiStat(CiStatNum) <= CiStatRg
--   CiStatNum <= SchBt(BtQoS(QoS))->Ci>>4
-- SchSt1E
--   CiStatRg <= CiStat(CiStatNum)
-- SchSt1F
--   CiStatRg(SchBt(BtQoS(QoS))->Ci(3:0))->Act <= 0
--   BTbl(ShClr) -- clear Bt shift enables
-- SchSt20
--   BTbl(ShLdRq,BtQoS(QoS)) & wait for BTbl(ShLdAk) -- set shift starting point
--   BTbl(ShDn)
--   SchBtRg->QoSCnt--
--   goto SchSt00
-- /**********************\
-- * Update Bubble Table *
-- \**********************/
-- SchSt21
--   BTbl(ShClr) -- clear Bt shift enables
```

```
--  if(SchBtRg->QoSCnt = 1)
--    goto SchSt00
--  BtNdx <= SchSrch(SchBtRg(BtQos(QoS)+1)->TtInt&->IvInt&-
>IvFrct,BtQos(Vbrrt))
--  if(BtNdx = BtQoS(QoS)
--    goto SchSt00
-- SchSt22
--  BTbl(ShLdRq,BtNdx) & wait for BTbl(ShLdAk) -- set shift starting point
--  BTbl(ShUp)
-- SchSt23
--  BTbl(ShClr) -- clear Bt shift enables
-- SchSt24
--  BTbl(ShLdRq,BtQos(QoS)) & wait for BTbl(ShLdAk) -- set shift starting point
--  BTbl(ShDn)
--  goto SchSt00
-- /****************\
-- * Send idle cell *
-- \****************/
-- SchSt25
--  Assert SwIdle
--  GetSr(SchPtRg->Pt)
--  if(PollQoS = Pcbr)
--    goto SchSt21
--  goto SchSt00
```


15

A Single Scheduler to Fully Service Multiple Qualities of Service

Description of Related Arts

Traditional ATM cell schedulers and QoS oriented packet schedulers implement separate logic blocks for each Quality of Service (QoS). These logic blocks output their cell/packet streams to individual queues or FIFO's which are then drained by another logic block prioritizing between the queues.

With this approach, there is considerable duplication of logic since the service for any QoS contains much core functionality common to the service of any other QoS. The existence of multiple separately managed queues or FIFO's also requires large amounts of storage, either in the form of register bits or RAM cells. Highly intelligent queue management can increase storage efficiency but adds complexity.

Figure 43 is a block diagram showing a prior art scheduler for multiple qualities of service.

Summary of the Invention

A limited set of functions is necessary to support the basic QoS's required and recommended in current standards issued by bodies such as the ITU-T, ATM Forum and IETF. A single scheduler can be implemented with algorithms to perform these basic functions and with access to control parameters allowing easy adaptation to future QoS's and Flow Control mechanisms.

Figure 44 is a block diagram showing a single scheduler to fully service multiple qualities of service according to the present invention.

The invention reduces complexity by providing a common core that performs the micro-scheduling operations and decisions to support constant, variable and unspecified bit rate services plus providing easily accessible primitives facilitating extension to future high level QoS, flow control and policy based flow management functions.

Detailed Description of the Invention

The invention places into a single scheduler the basic functionality to support multiple QoS's. A few key architectural considerations contribute to this capability.

1. A key is created for each flow containing the following information: QoS number in which the higher the QoS priority the lower the number; desired time, measured in transmit opportunities, for the next transmit; Interval time desired between transmits; flow number. Instantaneous transmit priority decisions among incoming flows are made based on the ordered set of these keys plus a partial index that identifies changes in QoS number.

2. QoS's may be assigned more than one number and the facility is provided to switch between the numbers dynamically in order to accommodate conditional changes in priority of the stream as window sizes or the guaranteed portion of its bandwidth in a time interval are met. It is important to note that the flow is not switched to a different queue but that the instantaneous priority of the data stream is changed. Thus there is no possibility of cell/packet order corruption as would be the case in a traditional queue/FIFO oriented implementation.

3. QoS numbers sufficiently large disable transmission to accommodate QoS's which require a temporary stoppage of cell/packet forwarding.

4. An interface is provided for external processes to dynamically change QoS number or Interval time to allow wide flexibility and expandability relative to high level QoS and policy management algorithms and logic.

Algorithm to Assign Scheduler Resource to Multiple Ports in Correct Proportions

Description of Related Arts

The dynamic variation of port speed in high bandwidth switches is a relatively new phenomenon. Analog modems have long had the feature of connecting at different bit rates, some of them even changing bit rates during a session. However, they operate over a standard 64 Kb voice channel through the network, simply changing their modulation scheme in adaptation to the signal to noise ratio they encounter in a given connection. Thus, even these variable bit rate sessions do not present the need, nor drive inventions to accommodate switching dynamic port rates.

The recent proliferation of ATM Bandwidth on Demand and Rate Adaptive DSL have injected a new characteristic into high speed switching, i.e. high speed ports whose line rates change dynamically.

The scheduling of cells/packets to an output port requires some amount of scheduler time for each cell/packet. The scheduling process bandwidth inherently becomes a critical resource. Every switch implementation faces the problem of managing this resource. Where throughput performance is the most crucial requirement, a dedicated scheduler is provided for each port so each port has immediate access to scheduling resource. However, when economics is important, scheduling resources are shared among multiple ports.

It becomes complicated when the scheduling resource is shared across ports with different line rates. It becomes particularly complicated when the ports that are sharing the resource have dynamically changing bandwidth needs.

Traditional algorithms for sharing the scheduler resource vary from simple round robin port service to weighted service based on an ordered list of port numbers. Round robin service works well when ports are approximately equal in line rate. Weighted priority works well when the port line rates are constant. Neither of these traditional solutions works well when port line rates vary dynamically. In this case, the implementation must suffer the expense of multiple schedulers or suffer performance degradation.

Summary of the Invention

This invention applies an algorithm to determine port service order which is similar to those used in QoS scheduling of cells/packets within a port. Depending on performance and cost requirements, the algorithm may be implemented in hardware or software.

The decision process to determine port sequencing order is equivalent to that used in determining which cell/packet to send next among the many flows of different rates passing through a single port. However, the frequency at which port sequencing decisions need to be made is typically two or three orders of magnitude less than the frequency that per-flow decisions need to be made. Thus it will usually be practical to perform port sequencing decisions in software or by allocating a very small percentage of hardware capacity from the existing cell/packet scheduler.

The result is efficient allocation of switching resources across ports even in the presence of dynamic port bandwidth changes with little or no added cost.

Detailed Description of the Invention

A cell/packet scheduler is sequentially dedicated to individual ports. For each port, it decides which flow's cell/packet will be transmitted next on the port. Ports with higher line rates need to have the scheduler's service more frequently than ports with lower line rates. This invention addresses the algorithm for deciding the sequence in which ports are serviced by the cell/packet scheduler to assure that each port receives the proportion and frequency of service commensurate with it's line rate.

For readability the term "cell" will be used in the remainder of this document instead of "cell/packet" but it will be understood that the algorithm applies to either cells or packets.

The invention involves a stack of entries, each of which contains a port number and a key based on pseudo time. The stack is ordered so that the next port to be serviced is at the front. After the port in the front entry is serviced, the pseudo time key is updated and the entry is reinserted into the stack appropriately to queue up its next service commensurate with its line rate. Any change in a port's line rate will automatically be taken into account the next time the port is serviced.

The concept of pseudo time is used to provide a relative numeric indicator of temporal need for service. Rate is also measured in normalized units. A sorted stack of pseudo times and port numbers is used to choose the next port to service.

A table, indexed by port number, is kept separately. Each entry in the table contains an interval count that correlates to the line rate of the port.

A range of numbers is needed in which to meaningfully represent relative line rates and service times. The top of this range is defined by adding the cells/packets per second for all ports at their maximum line rates and then multiplying this my a number sufficiently large to expand it by an order of magnitude, for example by 10 or 16. This increased number provides good granularity in representing relative rates and intervals.

By way of example, consider a system with N ports. The sum of the maximum cells per second of all the ports is calculated. This sum is multiplied by 10. For convenience in calculations, the result is rounded up to an even power of 2 and is stored in the variable named "Range".

The system is initialized as follows:

Each port is assigned a port number, "PortNum".

Each port has an initial line rate, "PtRate(PortNum)" in cells per second.

An array of per port service intervals, " Intvl (PortNum)" is initialized for all ports according to the initial line rates of the ports:

    Intvl (PortNum) = Range/PtRate(PortNum).

A service request stack entry, "SrvRq", is constructed for each port. SrvRq has three fields. " SrvRq->RqTime" contains the pseudo time count identifying when the port would like to be serviced next. "Stk->Intvl" contains the desired service interval. "SrvRq->Port" contains the port number asociated with the stack entry. These fields are initialized to the following values:

    SrvRq->RqTime = 0
    SrvRq->Intvl = Intvl(PortNum)

SrvRq->Port = PortNum

The service request stack entries are ordered on the stack by the key consisting of the first two fields with SrvRq->RqTime being most significant and SrvRq->Intvl being least significant. The entry with the lowest key is at the front of the stack. Initialization is now complete and service can begin.

Port service proceeds as follows:

The port whose number is in field SrvRq->Port of the SrvRq entry on the front of the stack is serviced. This SrvRq is then updated:

SrvRq->RqTime += Intvl(PortNum)
SrvRq->Intvl = Intvl(PortNum)

Note that the port rate, and therefore Intvl(PortNum) may change between services of a port. The contributing conditions and timing of such changes are specific to the port and not part of this algorithm. When port rate changes occur, the external process associated with operating the port will update Intvl(PortNum) and the new rate will automatically be incorporated into port service sequencing.

The SrvRq at the front of the stack is removed and its key is compared with the remaining entries on the stack. It is then inserted in the location its key dictates.

The front SrvRq on the stack now indicates the next port to be serviced.

It is appropriate to note that there are several methods to accomplish comparison and location of keys on the stack. Binary search and balanced tree indexing are two examples. The search/compare method is not part of this invention.

Beaded Buffer Pointer Chain With Intermediate Pointers

Description of Related Arts

Data switches have evolved through a number of stages. The early switches simply moved data into RAM from one port and forwarded it out from RAM to another port. Modern switches contain several processes such as protocol engines and interworking functions. Data parcels are not passed from process to process, rather they are stored once and pointers to the data is passed from process to process.

In order to support Quality of Service (QoS), multiple pointer queues are required. Queuing and dequeuing become a major portion of the overhead. Switch performance becomes highly dependent on queue architecture.

Multiple data queues combined with multiple processes acting on each data queue means at least two sets of interrelated queues with some type of links to each other. Each time a process touches a data parcel, one or more entries is queued and dequeued at least once and often more than once.

Some parcels are really pieces of larger parcels. Examples are ATM PDU's made of many ATM cells or fragmented IP packets or Fragmented Frame Relay frames. Processes that need to work on the larger parcels must either assemble them into new buffers or form another layer of pointers to track the larger demarkations.

Figure45 below illustrates prior art multiple queues associated with a buffer pool.

<u>Multiple Queues for Multiple Processes</u>

For each queue there is a pointer to the oldest and newest entry, often referred to as head and tail (or tail and head ... there is no consistency in common practice). Each entry contains a pointer to the next entry. Some architectures are bidirectional meaning that each entry contains pointers both to the next and to the previous entries.

Moving an entry from one queue to another requires updating a total at least four and sometimes six or eight pointers. Since large numbers of queues are involved, these pointers are often stored in RAM rather than registers so this amounts to considerable overhead.

The overhead of moving an entry from one queue to another can be understood by an example.

Assume two queues names Q1 and Q2. They have pointer in RAM addresses Q1_Oldest, Q1_Newest, Q2_Oldest and Q2_Newest. Moving an entry from Q1 to Q2 involves the following RAM accesses:

```
Tmp1 = Ram(Q1_Oldest)
Tmp2 = Ram(Q2_Newest)
Tmp3 = Ram(Tmp1)
Ram(Q1_Oldest) = Tmp3
Ram(Tmp2) = Tmp1
Ram(Q2_Newest) = Tmp1
```

Summary of the Invention

This invention is an improvement in queuing architecture. It combines multiple queues into a single queue, significantly decreasing the overhead as data parcels are passed from process to process.

The invention implements a queue with additional pointers. It contains oldest and newest pointers as in a traditional queue. Between these, it also contains first process, second process, etc. pointers.

When a process advances from one data parcel to the next, it doesn't dequeue and requeue, but only follows the chaining pointer to the next parcel. This reduces the overhead considerably.

These queues are organized on a per-flow basis. As such, they identify parcel sequencing order within the flow. This eliminates the need to reassemble larger parcels from smaller in a way that will be described later. This also reduces overhead and provides significant flexibility in process implementations.

Figure 46 below illustrates a single queue with pointers to service multiple processes, according to the present invention.

The overhead of advancing a process from one entry to the next can be understood by an example.

Assume one queue with an intermediate pointer at RAM address Q1_Proc3. Advancing an entry involves the following RAM accesses

```
Tmp1 = Ram(Q1_Proc3)
Tmp2 = Ram(Tmp1)
Ram(Q1_Proc3) = Tmp1
```

Detailed Description of the Invention

As background to understand the invention, the passage of a data parcel through a typical switch will be described.

Data parcels in a particular flow typically pass through the same ordered set of processes. The life of a data parcel within the switch begins when the parcel is received from a port and stored in a data buffer. Via pointer passing, it is handed to process#1 then process#2, etc. until it is finally sent out on a port and its data buffer released back to a free buffer pool.

When a data buffer is first allocated for storing a parcel, a pointer is removed from a free queue and put onto the service queue for the first process that will manipulate the parcel. When the first process is finished with the parcel, its pointer is removed from the first process queue and put onto the second process queue. This continues until the last process is finished with the parcel at which time the parcel is sent to an output port and its pointer is put back on the free queue, making it available for reuse with a new incoming parcel.

To understand the overhead of removing a pointer from one queue and putting it on another queue, it is necessary to understand queue structure.

The queue structure
A set of chaining pointers are associated with a buffer pool. There is a one to one correlation between chaining pointers and data buffers. The first chaining pointer is associated with the first data buffer and so on. Data parcels in a sequential flow will be stored in whatever data buffers are free. The corresponding chaining pointers link the parcels together in the proper sequence.

Two end-pointers are associated with each queue. These point to the oldest and newest pointers in the chain of buffers waiting for service.

Removing the oldest pointer from a queue begins with reading the end-pointer that points to the oldest buffer pointer. This value provides the address in the buffer pointer table from which to read the next pointer in the chain. The value read from this location is then written into the oldest end-pointer. Thus two memory reads and one memory write are required to remove a pointer from a queue.

Adding the pointer to another queue, on which it will become the newest pointer, consists of reading the end-pointer to the newest buffer. The address of the pointer to be added is stored in the location retrieved from the newest end-pointer. Then the address of the new pointer is written into the newest end-pointer. This amounts to one read and two writes..

In summary, each time a pointer is transferred from one queue to another, a total of three memory reads and three memory writes are performed.

The invention places intermediate process pointers between the newest end-pointer and the oldest end-pointer.

When a new pointer is moved from the free queue to the overall queue or from the overall queue to the free queue, it still requires three reads and three writes. The first time a process acts on a cell in such a queue, it will read its intermediate pointer and find it zero. It then has to read the oldest end-pointer and store that value in its intermediate pointer. This requires a total of 2 reads and 1 write. This is half the memory operations that would have been required to remove from one queue and add to another. On subsequent actions within the same queue, the intermediate pointer will be non-zero. In this case, the process reads the new pointer value from the address it read from its intermediate pointer. Again, there is a total of 3 memory accesses, half as many as would have been required to remove from one queue and add to another.

Thus, the overhead with a multiple pointer queue can represent nearly a 50% decrease over the traditional queue per process overhead.

The other important aspect of the invention is that the queues are per-flow. This means that all the data parcels in buffers for a particular flow are sequentially chained in the queue. The intermediate pointers tell how far along the chain each process has worked. It is no longer necessary for processes to reassemble parcels into super-parcels in order to have all the data of the super-parcel available. The nature of process that operate on super-parcels is that they are the last process to operate on the chain. They simply leave their parcels on the chain until the last parcel of the super-parcel shows up, then process from the oldest to their intermediate pointer, releasing pointers as they go. This reduces both overhead and complexity.

Fractional Interval Times for Fine Granularity Bandwidth Allocation

Description of Related Arts

Most cell scheduling algorithms contain a parameter which is the number of cell times between a particular VC's cells. For example a 64 Kbps VC running on a 2.048 Mbps line would occupy every 32nd cell slot so its interval time would be 32. The scheduler attempts to assign every 32nd cell time to this VC. If the cell slot is already taken by another VC, then a priority check is made. The higher priority QoS gets the slot and the lower priority VC gets bumped to the next free time slot.

Rates allowed by this mechanism are line rate, line rate /2, line rate /3, line rate/4, etc. The range of rates depends on the number of bits in the divisor. For example, a 10 bit divisor accommodates line rate down to 0.1% of line rate. At lower data rates, i.e. line rate over large divisors, the steps from one possible rate to the next are small. At higher rates, however, the steps are large. For example, the first programmable rate lower than line rate is 50% of line rate. The next smaller programmable rate under 50% line rate is 25% line rate. Smaller steps at the high end would be desireable.

Summary of the Invention

The invention defines interval times as fixed point fractional numbers, i.e. <u>each number has n bits of integer portion and m bits of fractional portion</u>. The number of integer and fraction bits depends on the range of rates and the desired granularity. For example, 10 integer bits plus 4 fractional bits allows a dynamic range of line rate down to 0.1% of line rate. The first step below line rate is 94% of line rate. Granularity rapidly improves as rates decrease. The step below 50% is 48.5%.

Detailed Description of the Invention

In the example implementation, for each flow there is defined a 16 bit non-integer interval time, "Intvl(FlowNum)." This breaks down into the most significant 12 bits which contain the integer portion, "Intvl(FlowNum)->Int," and the least significant 4 bits which contain the fractional portion, "Intvl(FlowNum)->Frct."

A port time, "PortTime", is maintained which is a count of cell time slots experienced by the port. This is a 14 bit integer.

Another 16 bit non-integer is maintained for each flow. This number, "TgtTime(FlowNum)" breaks down into a 14 bit integer field, "TgtTime(FlowNum)->Int" and a 4 bit fraction, "TgtTime(FlowNum)->frct."

<u>Initialization:</u>

When a flow is initially defined, it's Intv(FlowNum) is also defined. Its initial value of TgtTime(FlowNum) is calculated from Intv(FlowNum) and PortTime. Specifically:

TgtTime(FlowNum)->Int = PortTime + Intvl(FlowNum)->Int
TgtTime(FlowNum)->Frct = Intvl(FlowNum)->Frct

<u>Operation:</u>

Each time a cell slot passes for the port, TgtTime(FlowNum)->Int is compared to PortTime. (Actually, it is placed in queue ordered by TgtTime(FlowNum) and the first entry in the queue is compared to PortTime. When this flow's time and QoS priority arrive, it will find itself at the front of the queue and its time will be compared.) When PortTime becomes equal or greater than TgtTime(FlowNum)->Int then a cell from FlowNum is to be sent.

After its cell is sent, the target time for the next cell is calculated:

TgtTime(FlowNum) = TgtTime(FlowNum) + Intvl(FlowNum)

Note that this time the entire Intv, including both fractional and integer parts are added to the entire target time. This may or may not result in carry from the fractional to the integer portion.

The new target time is now ready to be compared once again to the port time, repeating the process beginning under Operation, above.

This simple process of accumulating the fractional buildup from the interval time into the target time automatically adjusts the repetition frequency for transmitting the flow's cells on the port to match the desired non integer rate for the flow.

One detail is not part of the invention but should be noted. As there is a finite number of bits, overflows may occur. Comparison logic obviously has to take this into account to maintain proper order in the target time queue and to make meaningful comparisons against port time.

Example:

As an example we consider a T1 transmission rate that corresponds to 1536 Kbps (Kilo Bits per second) of user data rate. If we choose to use 950 Kbps of this for our port cell flow.
1536 divide by 950 = approx. 1 5/8.

This translate into transmitting one ATM cell every 1 5/8transmit opportunities
If we add 1 5/8to itself sequentially we'll have the following sequences of fractional numbers:
0, 1 5/8, 3 1/4, 4 7/8, 6 ½. 8 1/8, 9 ¾, 11 3/8, etc.

The ATM data cells are transmitted at the integer portion of the above sequences, i.e. 0,1,3,4,6,8,9,11, etc. An ATM idle cell is inserted at the other intervals. The table below shows this data flow.
The PIC count (Port Interval Counter) keeps track of the following sequences

PIC	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
I			I			I		I			I		I			I			I	
D	D	D		D	D		D		D	D		D		D	D		D	D		D

I = Idel Cells
D= Data Cells

Multiple Preemptive CBR's for Precise Port Pacing Control

Description of Related Arts

There is sometimes a requirement to limit the effective bandwidth of an ATM port to a rate below the line rate. For example, in the case of a customer that is purchasing lower network bandwidth than the local loop is capable of carrying.

There are several ways of accomplishing this.

1. The network ingress port can police, i.e. discard any cells over the desired bandwidth. This restricts data coming into the network but has severe negative impact on the service seen by the customer. Data applications become extremely slow with even slight data loss. Discarding even a small percentage of cells renders the network service unusable for data applications. Voice and video quality suffers too, though not as severely as data applications.

2. Individual flows can be assigned peak rates with no overbooking. This is a way of limiting the total proffered data out of a port but is not a practical solution except in a few limited applications. Dedicating fixed bandwidth to data connections results in poor overall bandwidth utilization because they tend to be bursty. UBR service was conceived precisely so this application type could more effectively share the network pipe.

3. Add hardware to clock outgoing cells to a port at a lower rate than the port can forward them. This is not a particularly desirable solution because it adds synchronous time features to the switching function that would otherwise only be concerned with cell sequencing.

Solution number 3 is the most common.

When ports are limited in rate, i.e. throttled below their line rate, there is often an issue with rate granularity. The reason is that the switching function deals in cell slots. Using every cell slot yields line rate. Every other slot yields 1/2 line rate. The lower the throttled rate, the finer the achievable granularity. 5% or 10% or 15% are achievable but if 80% or 85% or 90% is desired, the close choices may be 50% and 100%.

Summary of the Invention

The invention employs from 0 to several CBR cell flows which contain idle cells instead of data. These idle cells are transmitted exactly as data cells would be so the clocking is a function of the port PHY rather than the switching function. The switching function is therefore not burdened with network clocking and synchronous scheduling but is only required to do what every switching function already does.

When this invention allocates bandwidth to a CBR flow of idle cells, it effectively subtracts the programmed peak rate of the idle stream from the line rate. This has the benefit of fine granularity at high remaining port bandwidth since in this case the amount being subtracted is a programmed value that is a small percentage of line rate.

Use of multiple idle CBR flows has the advantage of providing fine granularity across the entire bandwidth range.

Partitionable Page Shifter With Self Timing Xor Chain

Description of Related Arts

One of the most effective mechanisms to schedule data transmission is via a flow transmission request queue ordered by target time. This mechanism has its challenges, however. Inserting and deleting to/from an ordered list requires shifting parts of the list up or down which can create long access times and high overhead.

There are several approaches to this problem, ranging from very slow and hi overhead bubble up/down to more efficient block shifts. Block shifts are gate intensive but greatly increase performance.

Block shifting typically involves loading the right data into the shifter, shifting up or down and storing it back.

Another challenge is that time values in the list typically increase monotonically and eventually overflow. When this happens, values that should be large appear small and vice versa so the comparison function becomes very complex.

Summary of the Invention

The Partitionable Page Shifter is a key component in schedule queue processing. It is the component that ordered queue entries up and down in order to insert or delete an entry. This invention provides features that significantly improve the performance of processing ordered scheduler request queues.

The first feature is the ability to partition the queue into multiple segments that can then be shifted up or down during a single clock. This feature enables block shifting to take place without loading and unloading partial sections into the block shifter.

The second feature is the ability to efficiently determine the amount of time needed for logic to settle before performing the shift. The logic chain involved in defining shift partitions includes a string of XOR gates. The invention includes a mechanism for determining when control has propogated through the chain and is ready for the shift operation.

The third feature is the ability in a single clock cycle to adjust all queue entries for target time wrap. With this feature, time counters stay in a valid range for comparisons.

Detailed Description of the Invention

The first feature of the invention is associated with shifting entries up and down. There are two unique aspects to this feature. The first is the ability to partition the list into shifted and non-shifted areas. The second is avoiding excess waiting for partition control logic to propogate through the system before performing the shift.

The Partitionable Page shifter contains ordered entries. Each entry contains a time value and other information. The entries are ordered by time value. Time value fields contain at least more bits than the largest increment that will be applied to them, assuring that any wrap that takes place can not affect more than the most significant bit of the time field.

Each entry is held in a register that can be parallel loaded or loaded from the register above or below it. Thus, any register can be loaded, upshifted, downshifted or held unchanged.

Shifting is enabled by a signal that is conditionally propogated from the first to the last entry. An addressable flip flop is associated with each entry As shift enable is propogated from one entry to the next, it is inverted or not inverted depending on the status of the flip flop associated with the entry.

As there may be a large number of entries, the shift enable propogation might be quite lengthy. Rather than waiting the maximum time for this propogation as would be the traditional approach, a mechanism is employed wherein the enable value at the addressed flip flop and of the end of chain are sampled when a flip flop is set. The enable signal is monitored so that when the new signal reaches the end of the chain, a ready signal is generated, potentially allowing the shift to occur long before a maximul full propogation plus conservative heardroom would allow.

A shift operation is performed as follows:

1. All flip flops are cleared.
2. One or more flip flops are addressed and set. As a result, shifting is disabled from before the first entry to the first set flip flop, then enabled from the first to the second set flip flop then disabled from the second set flip flop to the third and so on.
3. When the ready signal appears, an up or down shift takes place in a single clock cycle and only within the enabled blocks.

The following are third party software building blocks used in the Mako-Dexter 3000:

1) TELOGY, Germantown, MD:

- Basic Access Switched Mode Client uP Component with TSGM and ISDM
- AAL2 DSP with G.711 / 729AB / 726 / 727 voice codecs and std. fax relay (also supports Telogy T3 proprietary encaps.)

2) INTEGRATED SYSTEMS / WIND RIVER SYSTEMS, Alameda, CA:

- pSOS Single Processor Operating System

3) INVERNESS SYSTEMS LTD., Kfar Saba, Israel (Virata):

- AAL0 / AAL5 drivers
- AF-VTOA-0075.000 CES Interworking
- MPC860 drivers and system services

4) TELENETWORKS (Next Level Communications, Rohnert Park, CA):

- ISDN BRI ETSI Net3 and DL Core (Network Side)
- ISDN BRI QSIG Layer III
- ISDN PRI ETSI Net5 and DL Core (Network Side)
- ISDN PRI QSIG Layer III
- Q.SAAL

Express Mail #EJ130608565US
Attorney Docket M015-1001

1/27

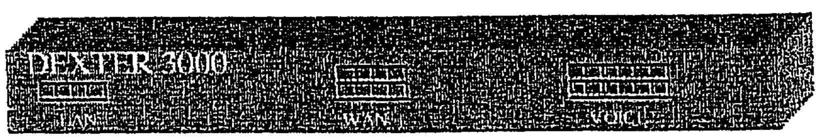


Fig 1

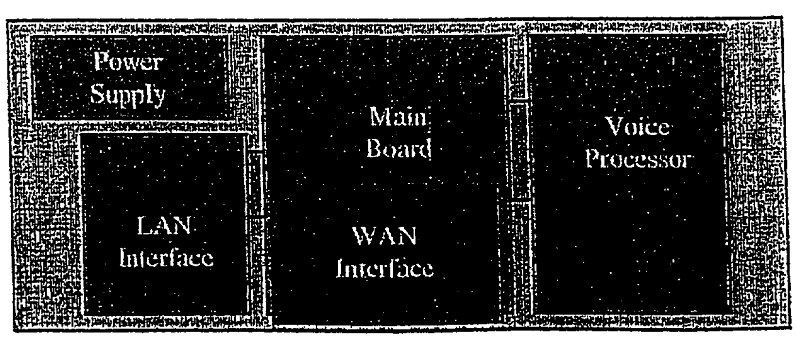


Fig 2

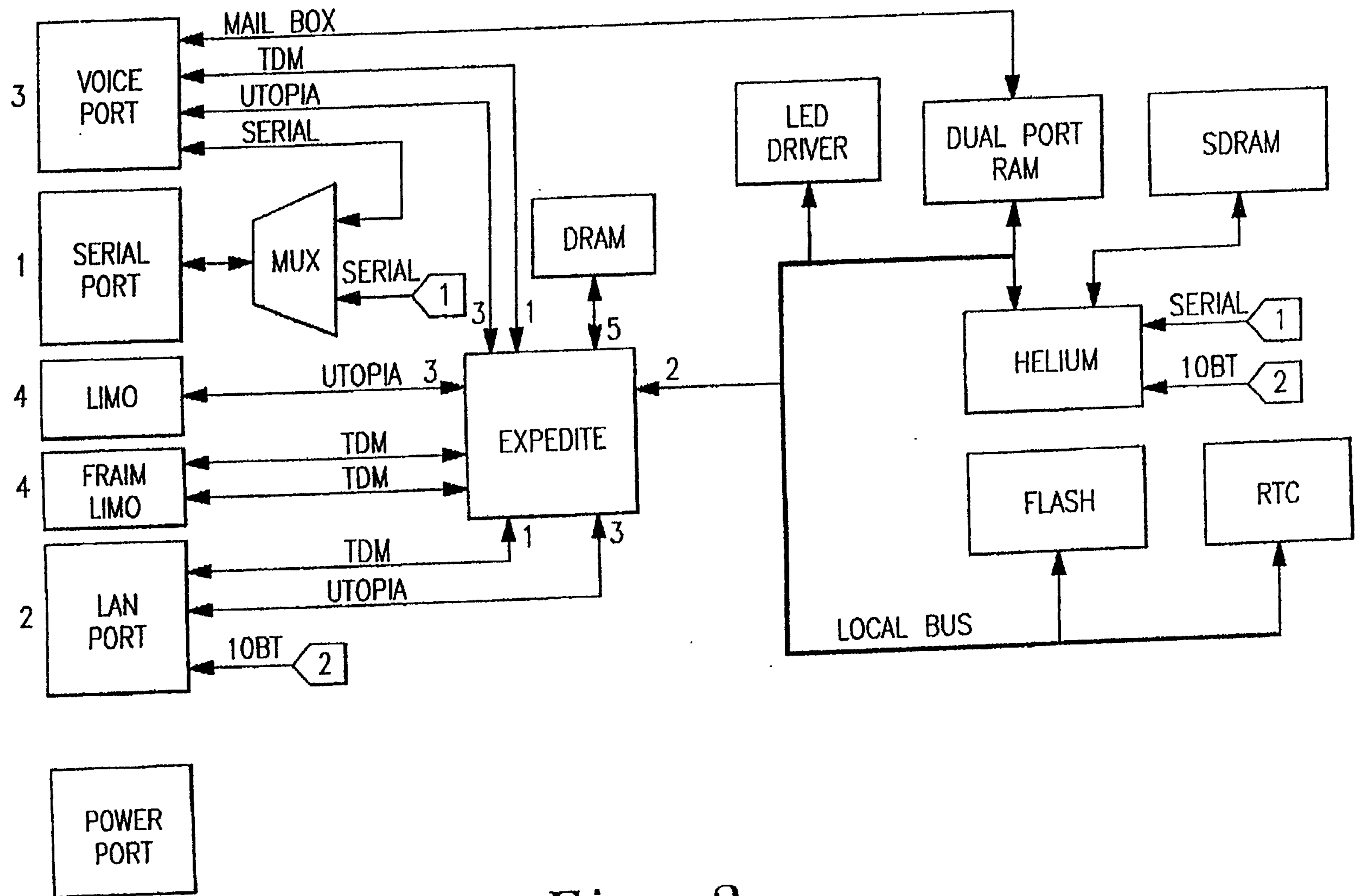

VOICE PORT
3
SERIAL PORT
1
LIMO
4
FRAIM LIMO
4
LAN PORT
2
POWER PORT
MAIL BOX
TDM
UTOPIA
SERIAL
MUX
SERIAL
1
UTOPIA 3
TDM
TDM
TDM
UTOPIA
10BT
2
DRAM
5
EXPEDITE
3
1
1
3
2
LED DRIVER
DUAL PORT RAM
SDRAM
HELIUM
SERIAL
1
10BT
2
FLASH
RTC
LOCAL BUS

Fig. 3

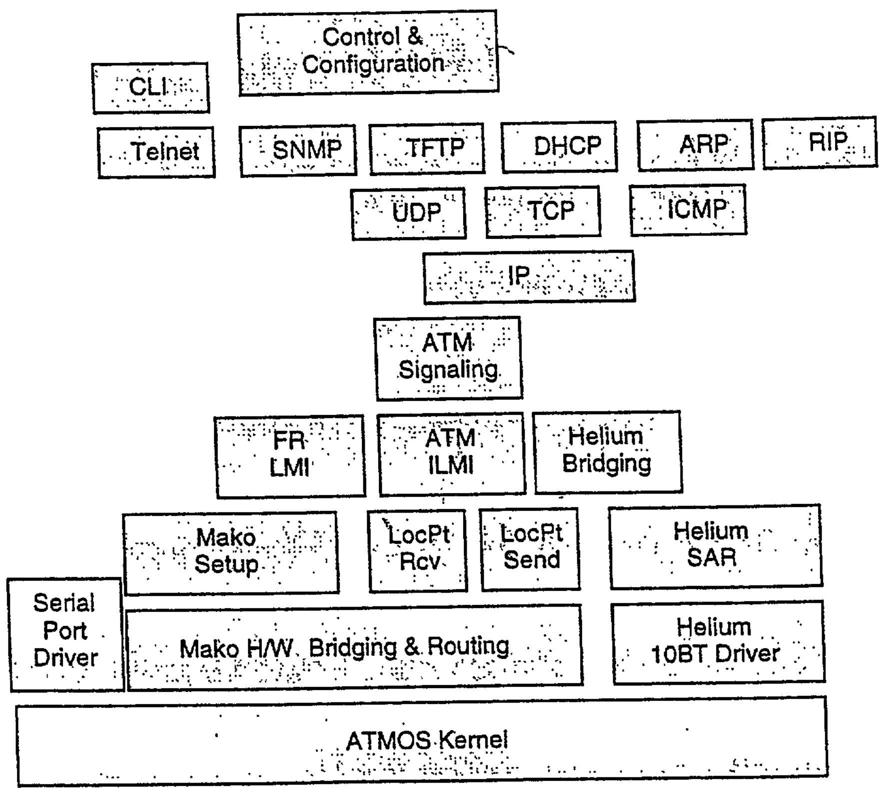

Control & Configuration
CLI
Telnet
SNMP
TFTP
DHCP
ARP
RIP
UDP
TCP
ICMP
IP
ATM Signaling
FR LMI
ATM ILMI
Helium Bridging
Mako Setup
LocPt Rcv
LocPt Send
Helium SAR
Serial Port Driver
Mako H/W Bridging & Routing
Helium 10BT Driver
ATMOS Kernel

Fig 4

4/27

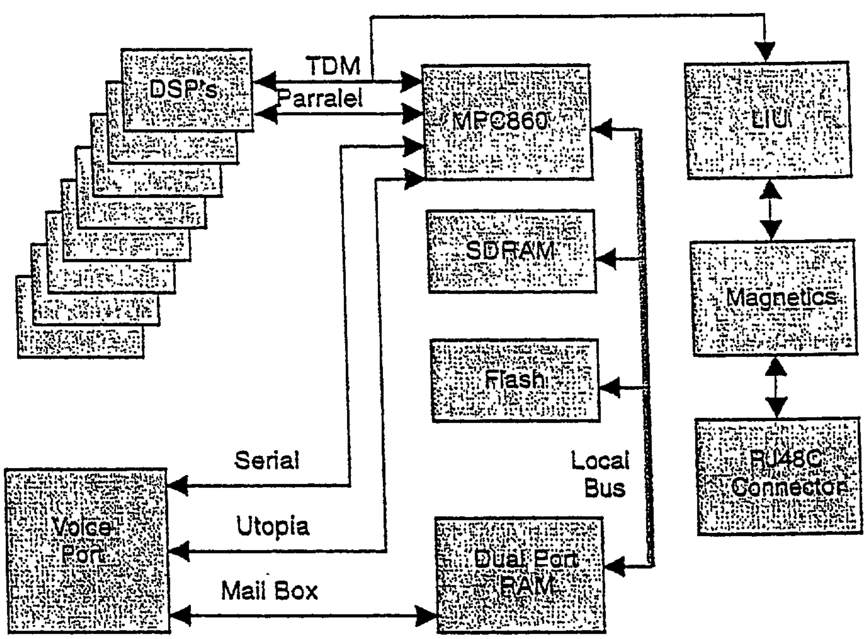
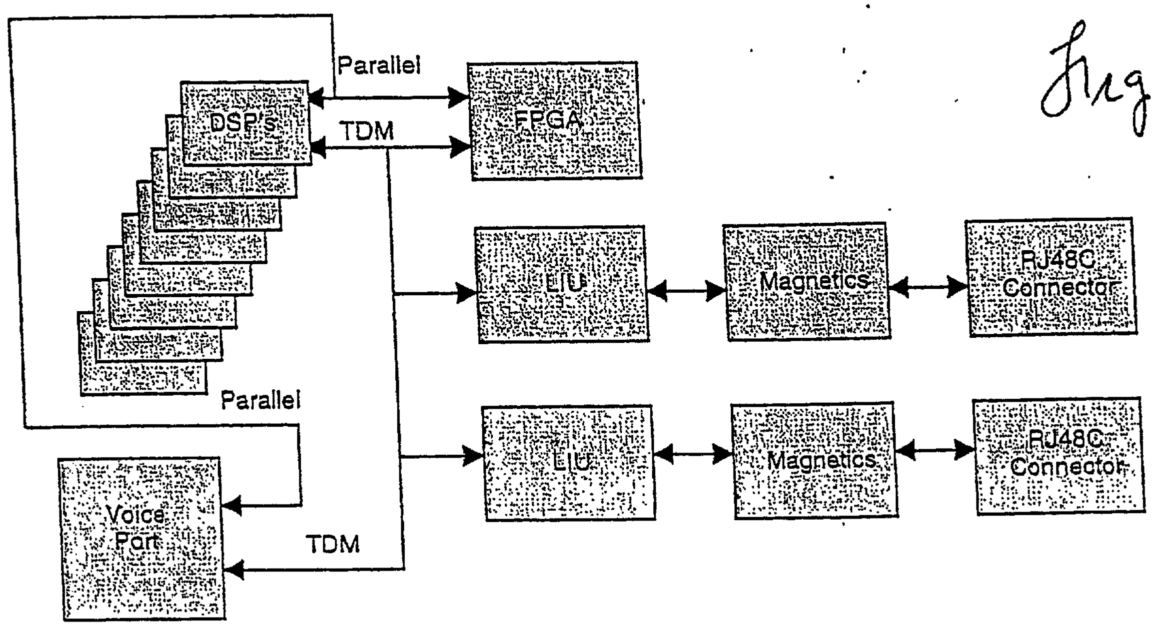

DSP's
TDM
Parralel
MPC860
LIU
SDRAM
Magnetics
Flash
RJ48C Connector
Serial
Local Bus
Utopia
Voice Port
Mail Box
Dual Port RAM

Fig 5



Parallel
DSP's
TDM
FPGA
LIU
Magnetics
RJ48C Connector
LIU
Magnetics
RJ48C Connector
Parallel
Voice Port
TDM

Fig. 6

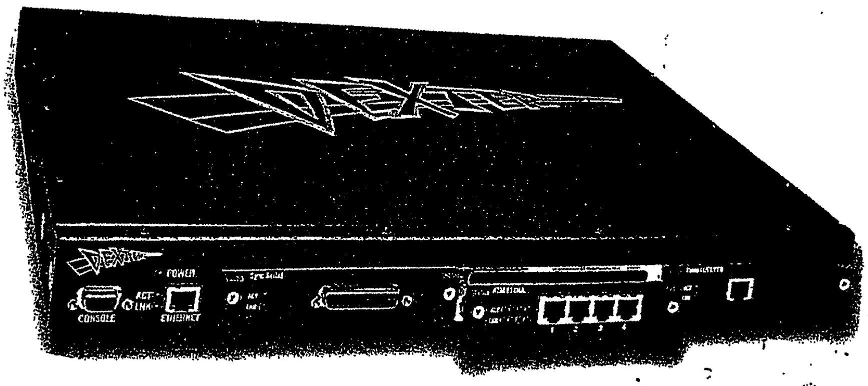


Fig. 7

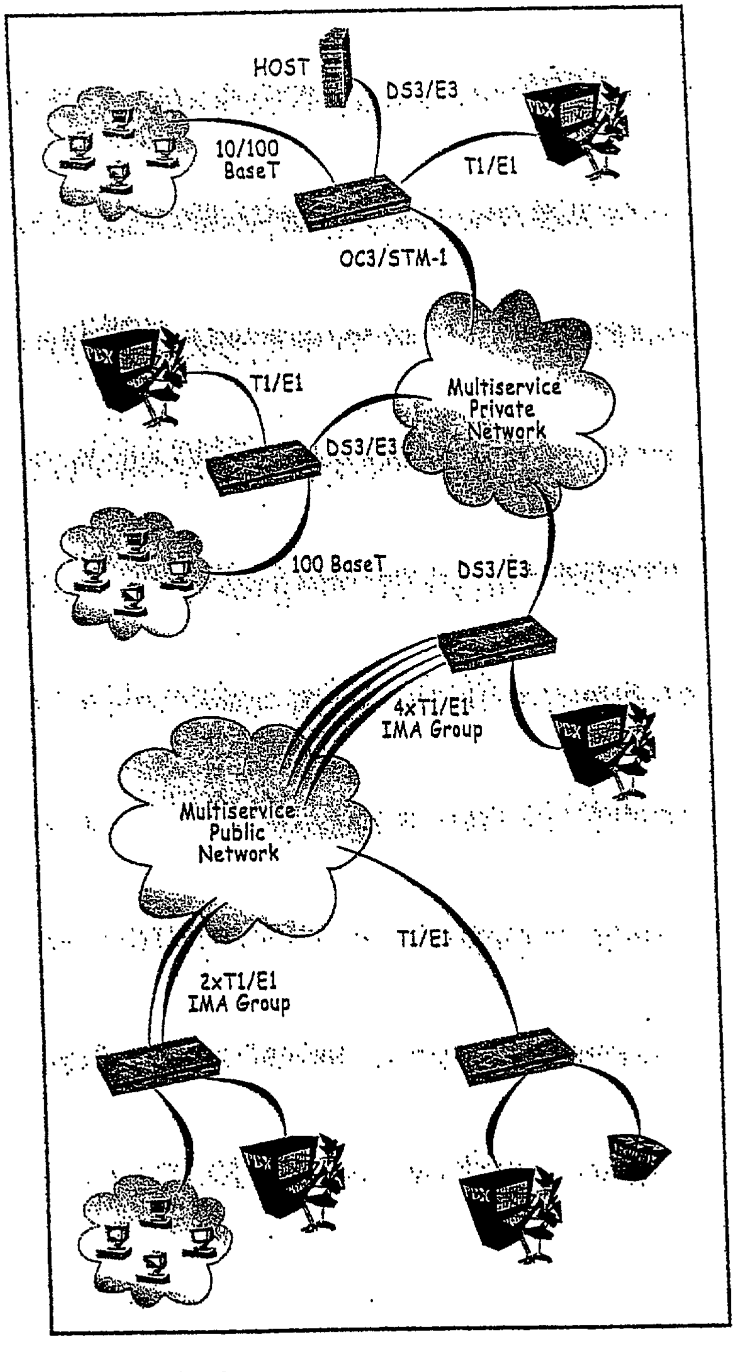

HOST
DS3/E3
10/100
BaseT
T1/E1
OC3/STM-1
T1/E1
Multiservice
Private
Network
DS3/E3
100 BaseT
DS3/E3
4xT1/E1
IMA Group
Multiservice
Public
Network
T1/E1
2xT1/E1
IMA Group

Fig 8

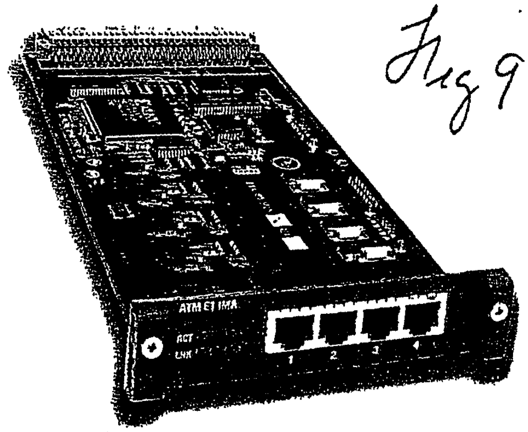


Fig 9

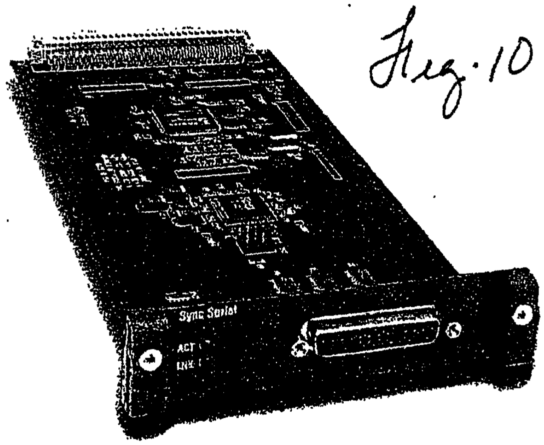
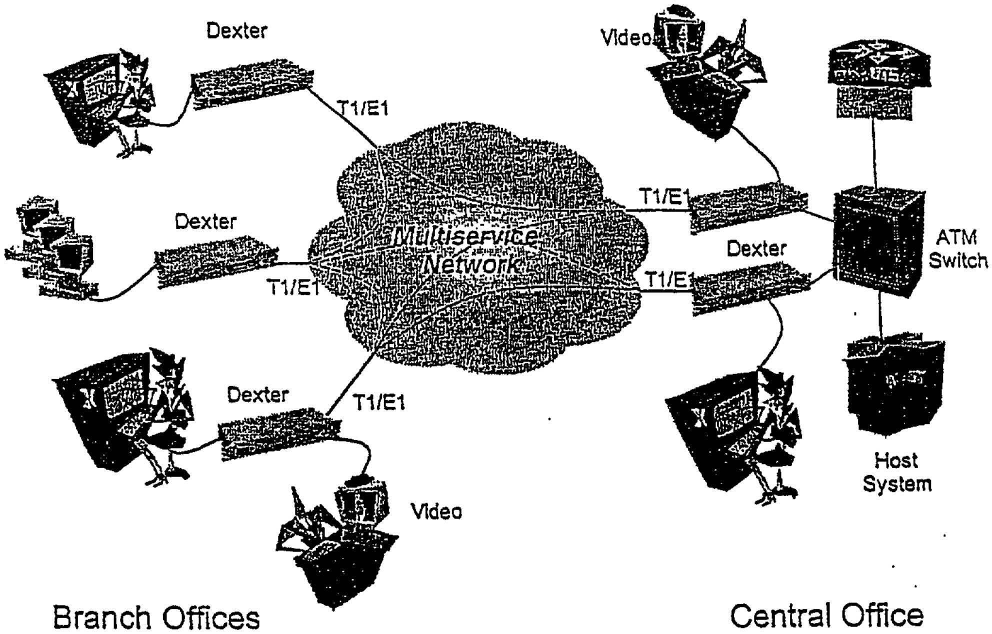


Fig. 10

Fig 11



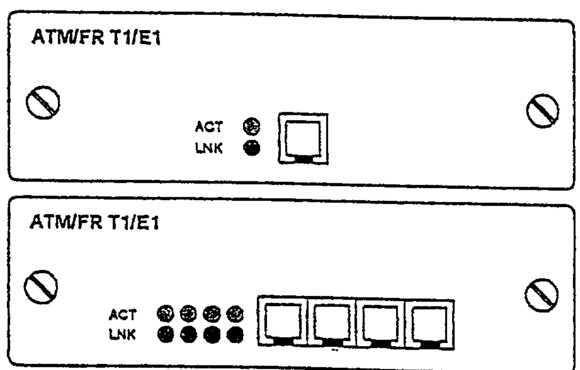


Fig. 12

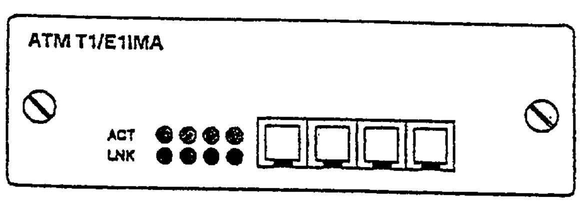


Fig. 13

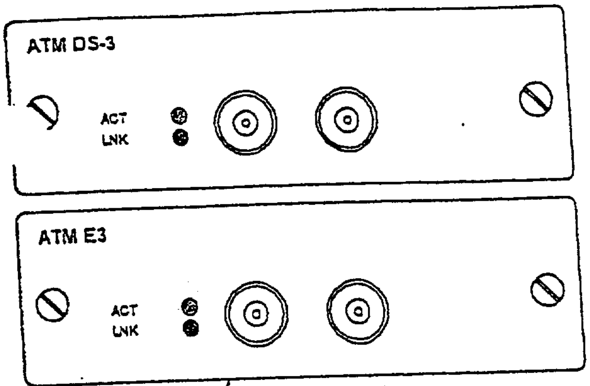


Fig. 14

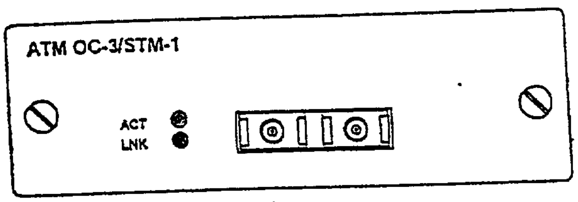


Fig 15

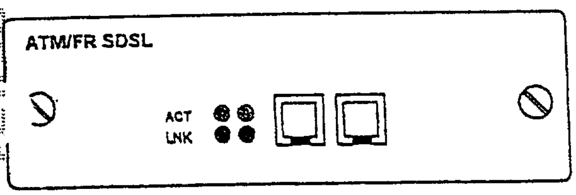


Fig. 16

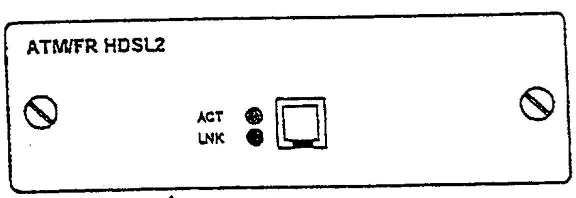


Fig. 17

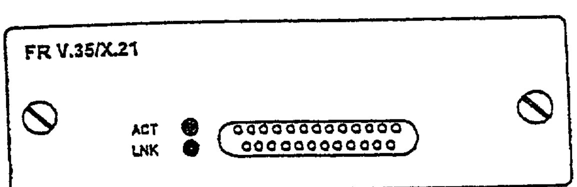


Fig. 18

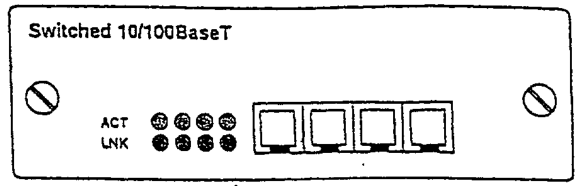


Fig. 19

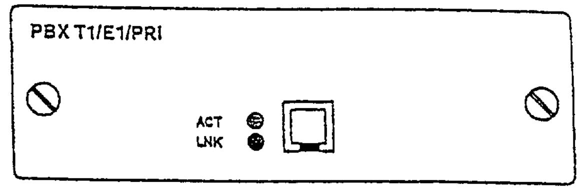


Fig. 20

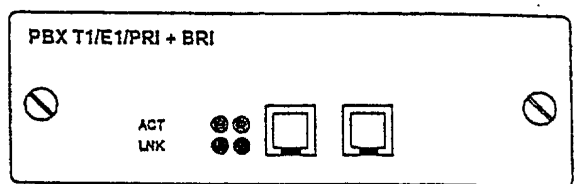


Fig. 21

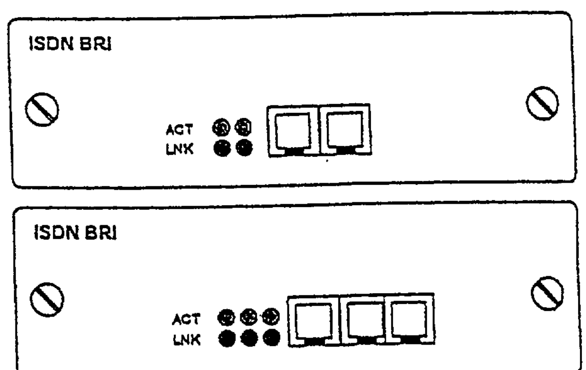


Fig. 22

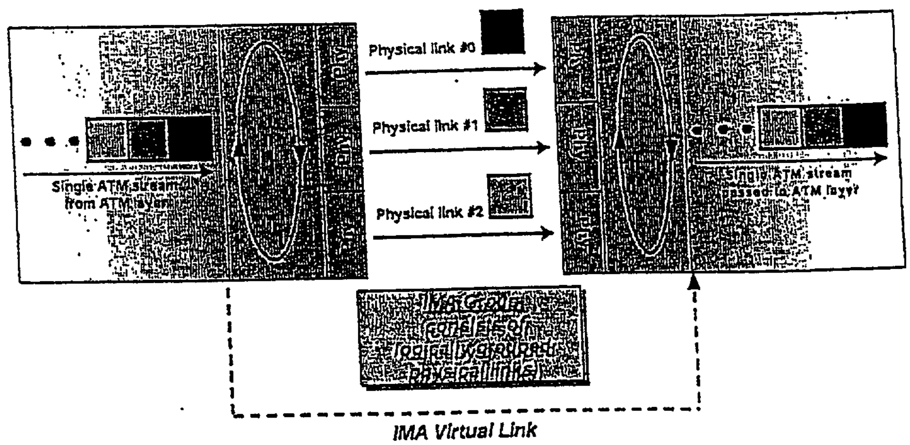

Physical link #0
Physical link #1
Physical link #2
Single ATM stream from ATM layer
Single ATM stream passed to ATM layer
IMA Virtual Link

Fig 23

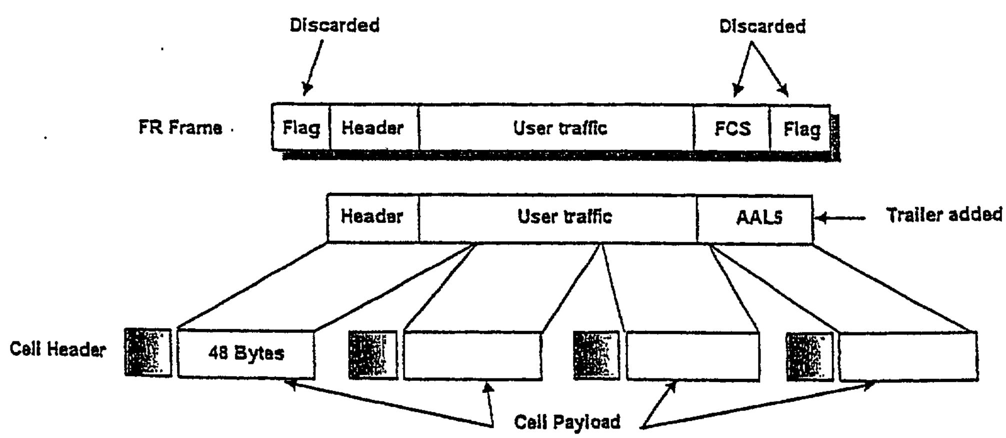

Discarded
Discarded
FR Frame
Flag
Header
User traffic
FCS
Flag
Header
User traffic
AAL5
Trailer added
Cell Header
48 Bytes
Cell Payload

Fig. 24

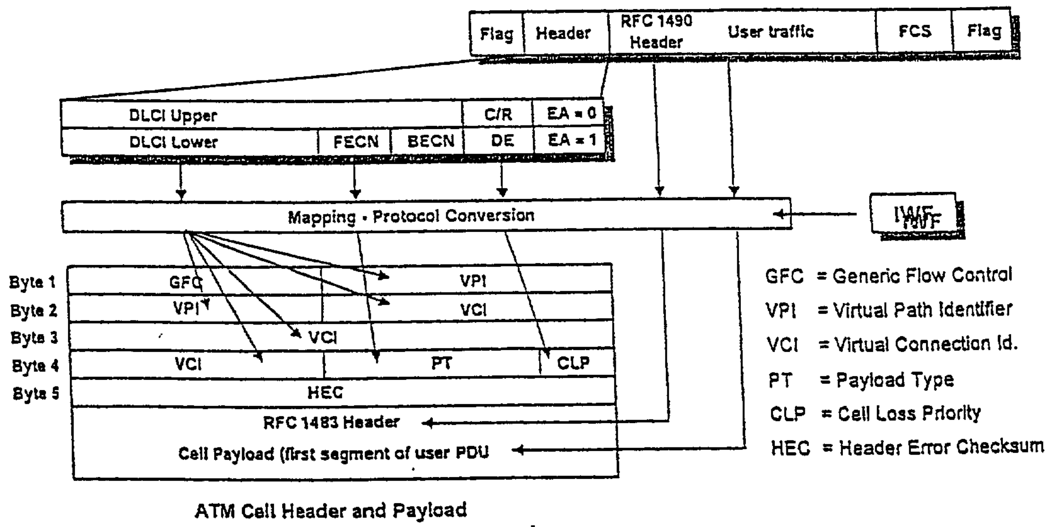

Flag
Header
RFC 1490 Header
User traffic
FCS
Flag
DLCI Upper
C/R
EA = 0
DLCI Lower
FECN
BECN
DE
EA = 1
Mapping - Protocol Conversion
IWF
Byte 1
Byte 2
Byte 3
Byte 4
Byte 5
GFC
VPI
VPI
VCI
VCI
VCI
PT
CLP
HEC
RFC 1483 Header
Cell Payload (first segment of user PDU
GFC = Generic Flow Control
VPI = Virtual Path Identifier
VCI = Virtual Connection Id.
PT = Payload Type
CLP = Cell Loss Priority
HEC = Header Error Checksum
ATM Cell Header and Payload

Fig. 25

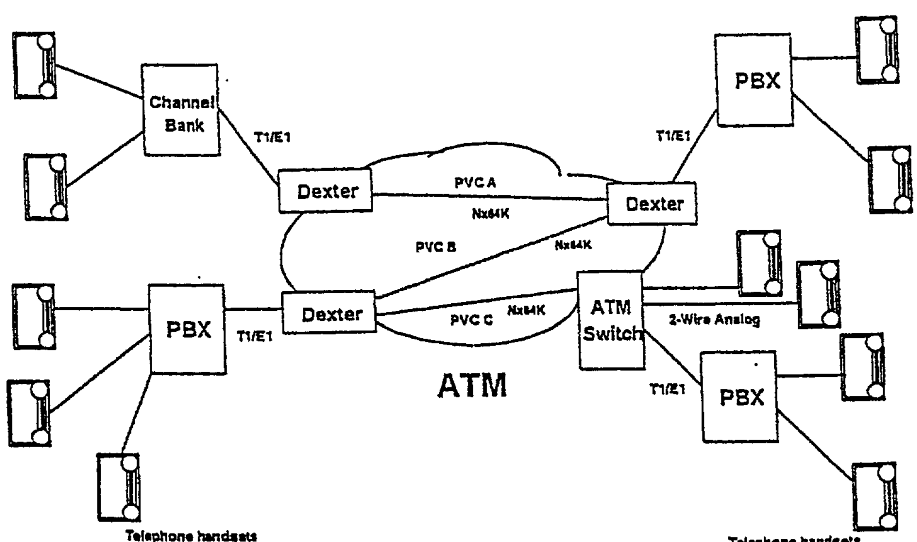

Channel Bank
T1/E1
Dexter
PVC A
Nx64K
PBX
T1/E1
Dexter
PVC B
Nx64K
PBX
T1/E1
Dexter
PVC C
Nx64K
ATM Switch
2-Wire Analog
T1/E1
PBX
ATM
Telephone handsets
Telephone handsets

Fig. 26

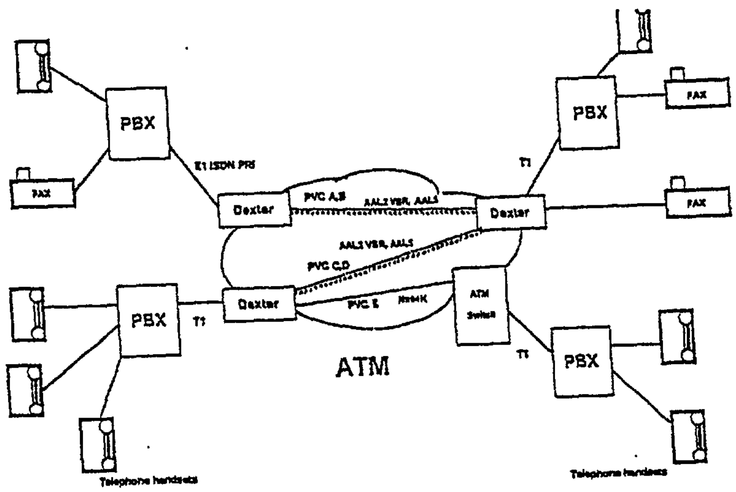

PBX
FAX
E1 ISDN PRI
Dexter
PVC A,B
AAL2 VBR, AAL5
T1
PBX
FAX
Dexter
FAX
AAL2 VBR, AAL5
PVC C,D
PBX
Dexter
PVC E
Nx64K
ATM Switch
T1
PBX
ATM
T1
Telephone handsets
Telephone handsets

Fig. 27

T08290"TEh96860

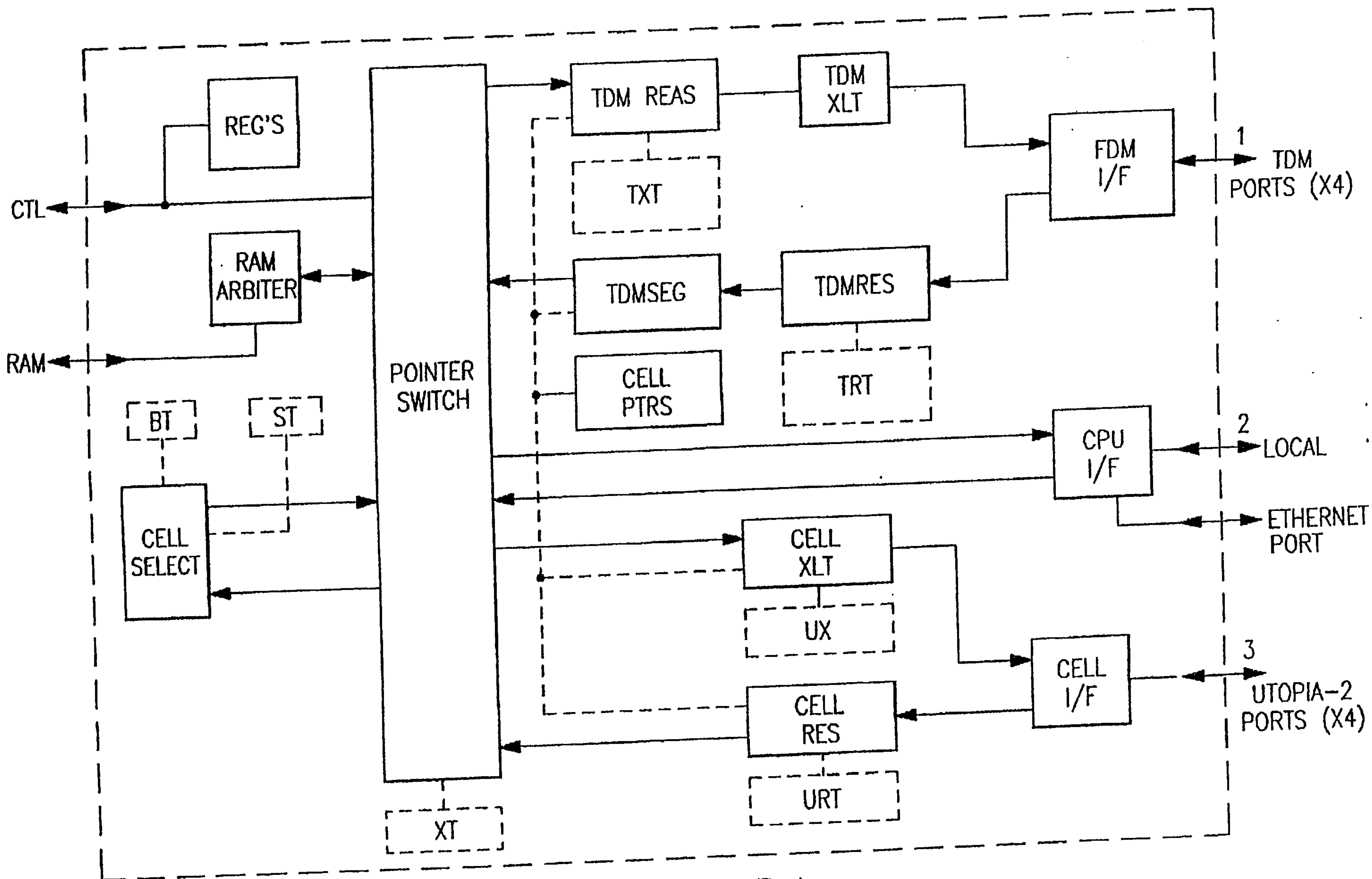


Fig. 28A

14/27

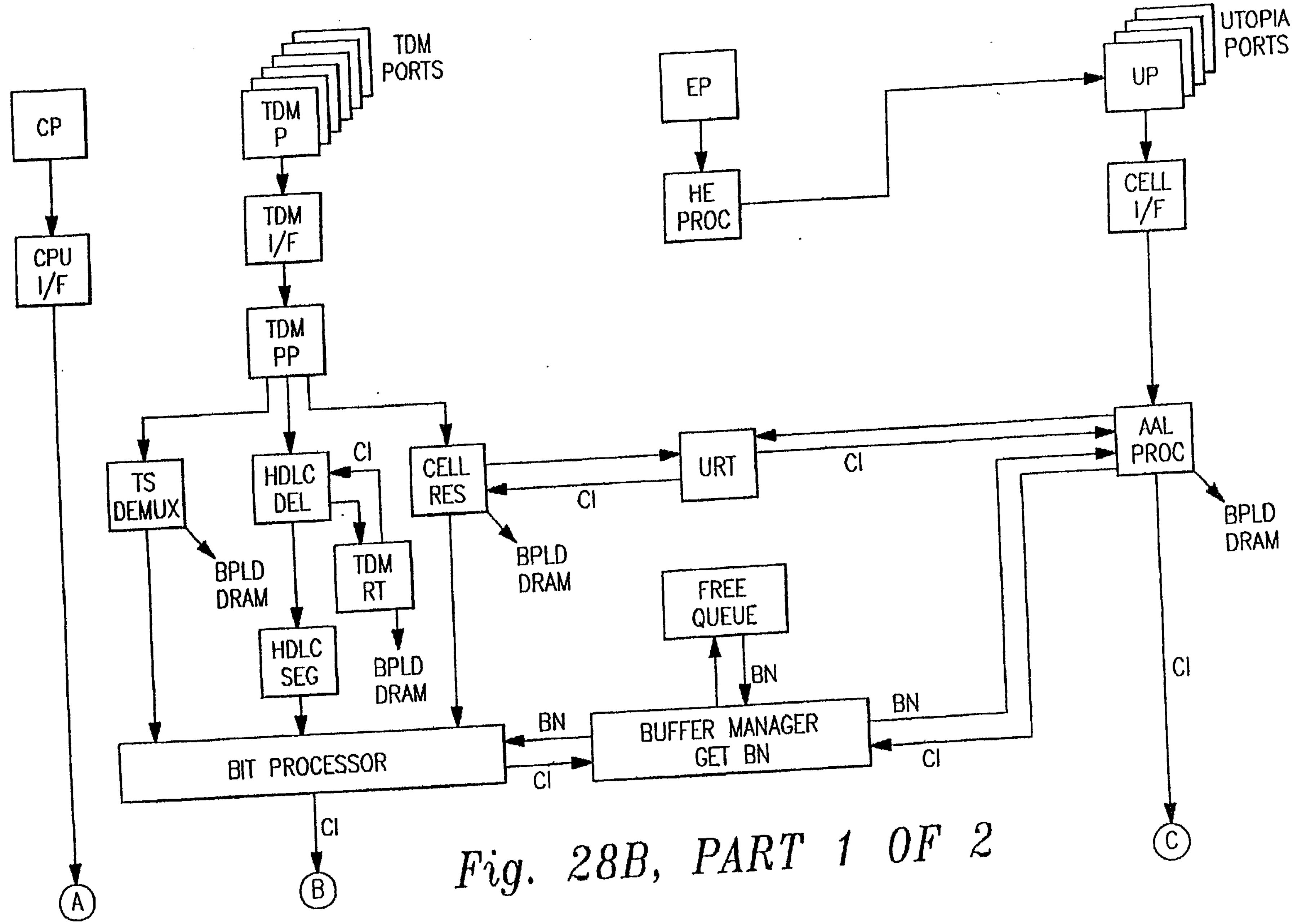


Fig. 28B, PART 1 OF 2

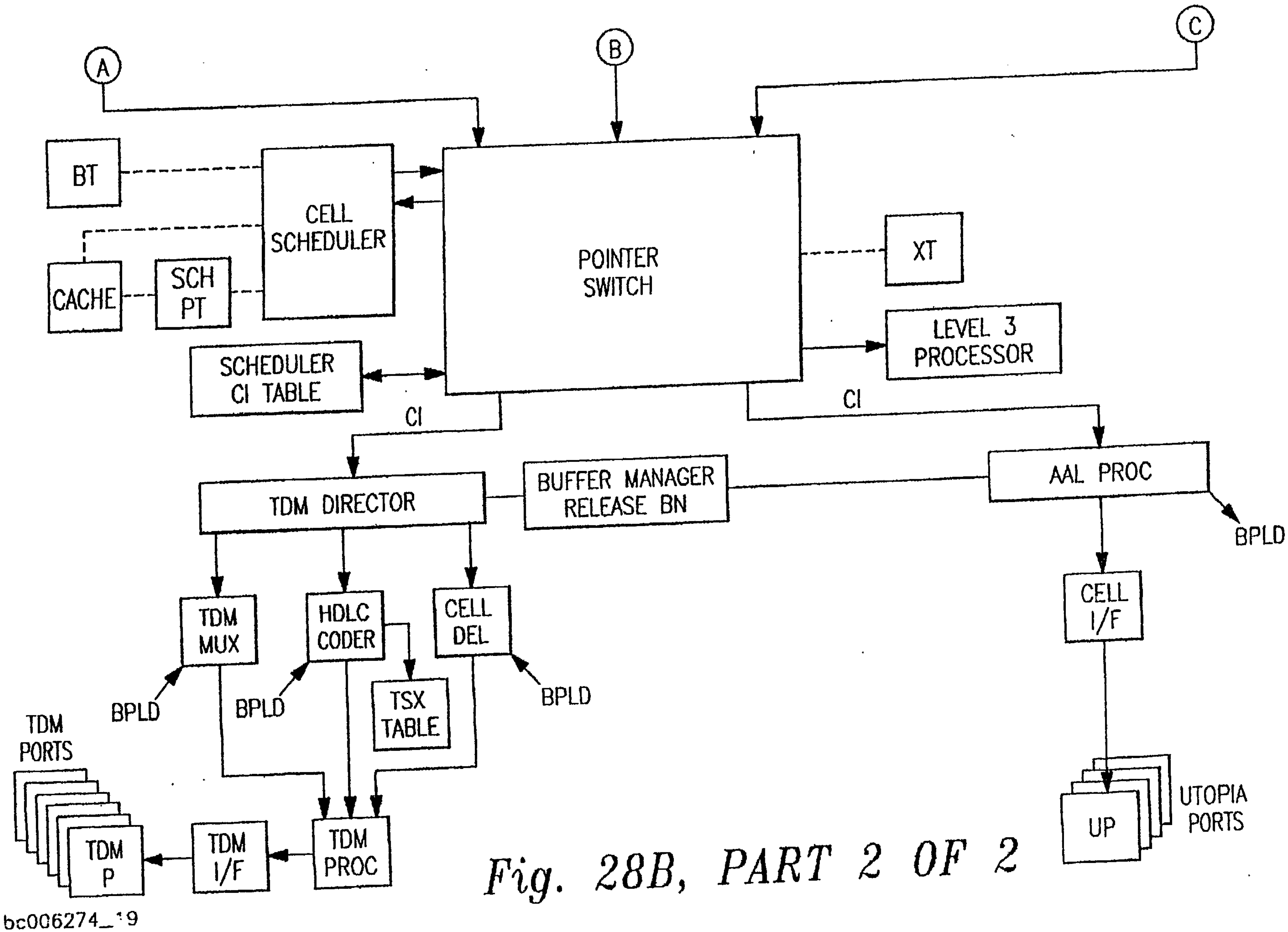


Fig. 28B, PART 2 OF 2

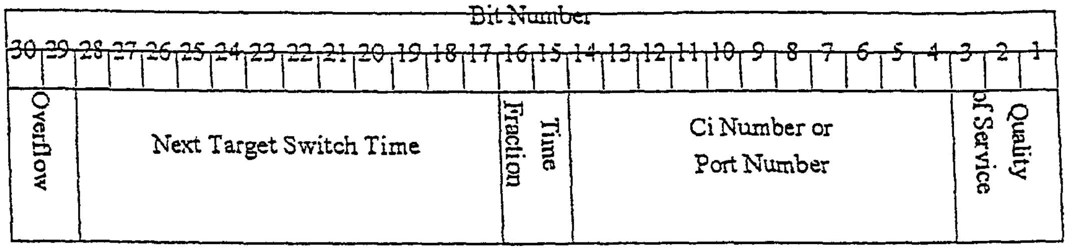
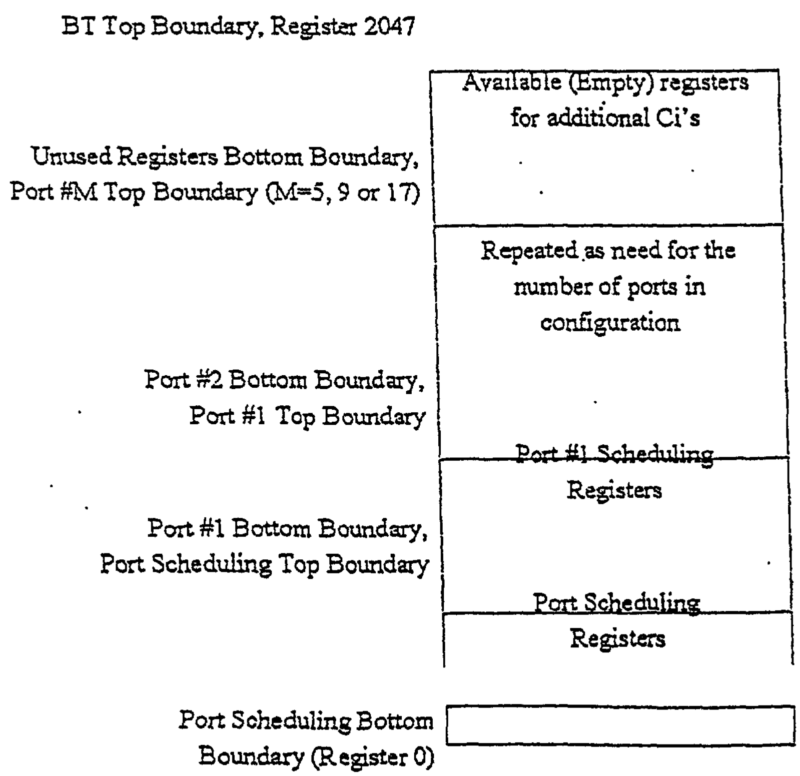


Fig. 29



Top Most Level BT Configuration

Fig. 30

Initial BT Location	Initial Target Switch Time	Port Number	Interval Value to be added for next Switch Time
4	8	5	8
3	6	4	8
2	4	3	8
1	2	2	8
0	1	1	2

Initial Port Settings with 4 TDM Ports

Initial BT Location	Initial Target Switch Time	Port Number	Interval Value to be added for next Switch Time
8	16	9	16
7	14	8	16
6	12	7	16
5	10	6	16
4	8	5	16
3	6	4	16
2	4	3	16
1	2	2	16
0	1	1	2

Initial Port Settings with 8 TDM Ports

Initial BT Location	Initial Target Switch Time	Port Number	Interval Value to be added for next Switch Time
16	32	17	32
15	30	16	32
14	28	15	32
13	26	14	32
12	24	13	32
11	22	12	32
10	20	11	32
9	18	10	32
8	16	9	32
7	14	8	32
6	12	7	32
5	10	6	32
4	8	5	32
3	6	4	32
2	4	3	32
1	2	2	32
0	1	1	2

Initial Port Settings with 16 TDM Ports

Fig. 31

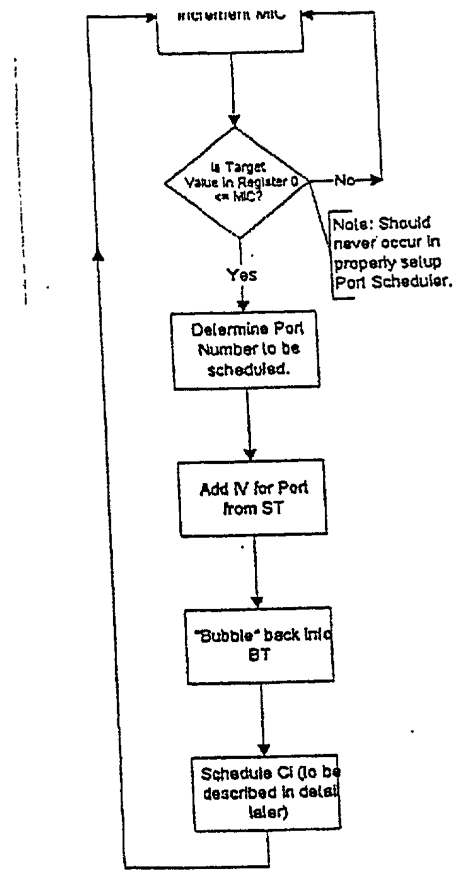


Fig. 32

Fig. 33

Register Position	Port Number	Target Time		Port Number	Target Time		Port Number	Target Time		Port Number	Target Time		Port Number	Target Time		Port Number	Target Time		Port Number	Target Time		Port Number	Target Time		Port Number	Target Time	
4	5	8		5	8		2	10		2	10		3	12		3	12		4	14		4	14		5	16	
3	4	6		4	6		5	8		5	8		2	10		2	10		3	12		3	12		4	14	
2	3	4		3	4		4	6		4	6		5	8		5	8		2	10		2	10		3	12	
1	2	2		1	3		3	4		1	5		4	6		1	7		5	8		1	9		2	10	
0	1	1		2	3		1	3		3	[illegible]		1	5		4	6		1	7		5	8		1	9	
	MIC incr to 1			MIC incr to 2			MIC incr to 3			MIC incr to 4			MIC incr to 5			MIC incr to 6			MIC incr to 7			MIC incr to 8			MIC incr to 9		
	Cell from Port 1			Cell from Port 2			Cell from Port 1			Cell from Port 3			Cell from Port 1			Cell from Port 4			Cell from Port 1			Cell from Port 5			Cell from Port 1		

-portant items to note from this example: The Ports are always in Target Time (TT) order. When the IV

Byte #(s)	9					8	7	5-6	3-4	1-2	
Bit #(s)	71-72	70	69	68	65-67	57-64	49-56	33-48	17-32	15-16	1-14
					QoS						Interval Value
Description	Future Use	No Cell Available	Ci Deactivated	Ci in BT		Cumulative Consecutive Empty Cells	Maximum Consecutive Empty Cells	Cumulative Count per Time Measure	Maximum Count per Time Measure	Future Use	in 12.2 format

Fig. 34

Ci ACTIVATION PROCESS

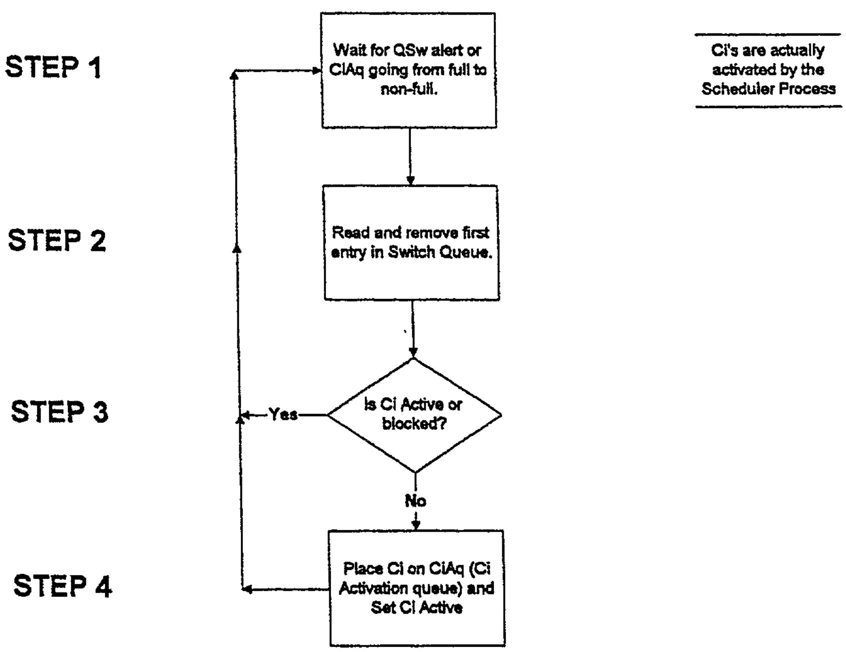


Fig. 35

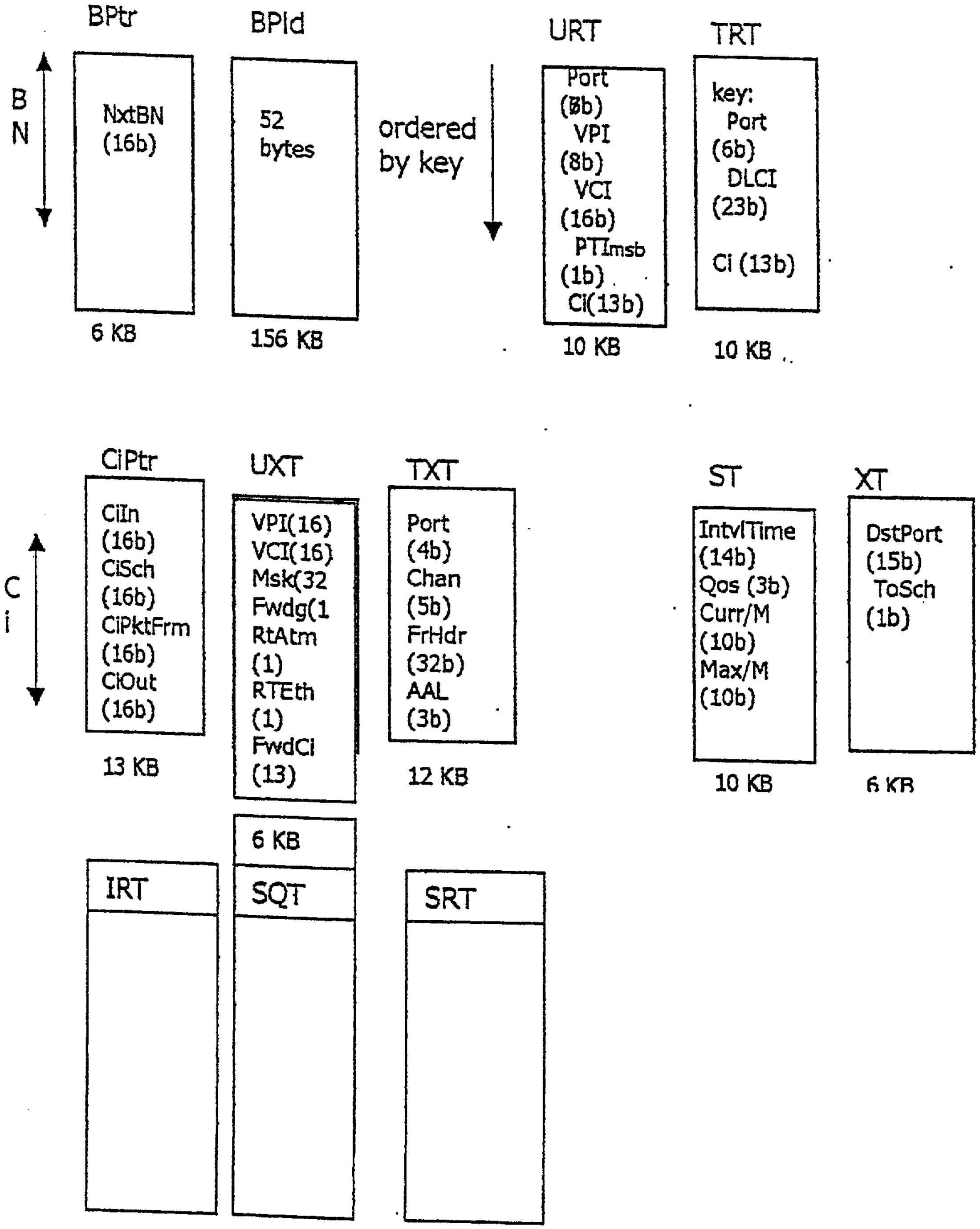

BPtr
BN
NxtBN
(16b)
6 KB
BPld
52
bytes
156 KB
ordered
by key
URT
Port
(8b)
VPI
(8b)
VCI
(16b)
PTImsb
(1b)
Ci(13b)
10 KB
TRT
key:
Port
(6b)
DLCI
(23b)
Ci (13b)
10 KB
CiPtr
Ci
CiIn
(16b)
CiSch
(16b)
CiPktFrm
(16b)
CiOut
(16b)
13 KB
UXT
VPI(16)
VCI(16)
Msk(32
Fwdg(1
RtAtm
(1)
RTEth
(1)
FwdCi
(13)
6 KB
TXT
Port
(4b)
Chan
(5b)
FrHdr
(32b)
AAL
(3b)
12 KB
ST
IntvlTime
(14b)
Qos (3b)
Curr/M
(10b)
Max/M
(10b)
10 KB
XT
DstPort
(15b)
ToSch
(1b)
6 KB
IRT
SQT
SRT

Fig. 36

Port Numbers	Port Type		Client Number	Client Type
0	Scheduler		0	CPU
1 – 64	Utopia PHY		1	ATM
65 – 1022	TDM		2	TDM
1023	Local Port		3,4	Scheduler

Fig. 37

Table	Register	Width	Estimated Size
URT	UrtBase	44 b	10 KB
TRT	TrtBase	42 b	10 KB

Tables indexed by Ci

Table	Register	Width	Estimated Size
CiPtr	CiPtrBase	64 b	10 KB
UXT	UxtBase	28 b	6 KB
TXT	TxtBase	44 b	12 KB
ST	StBase	51 b	10 KB
XT	XtBase	24b	6 KB

Tables indexed by Channel (Port) Number

TST	TstBase	76 b	5 KB

Tables indexed by BN

Table	Register	Width	Estimated Size
BPtr	BptrBase	12	6 KB
BPId	BpldBase	52	156 KB

Fig. 38

FIFO

The FIFO are stored in off-chip RAM. The FIFO sizes will be decided after simulation.

FIFO	Width	Estimated Size
TrmOut	16b	TBD
PktOut	16b	TBD
CellOut	16b	TBD
LocalOut	16b	TBD

Fig. 39

Bit	Description
0	Unknown frame count >= 128
1	Bad frame count >= 128
2	Overflow frame count >= 128
3	Unknown cell count >= 128
4	Bad cell count >= 128
5	Overflow cell count >= 128

Fig. 40

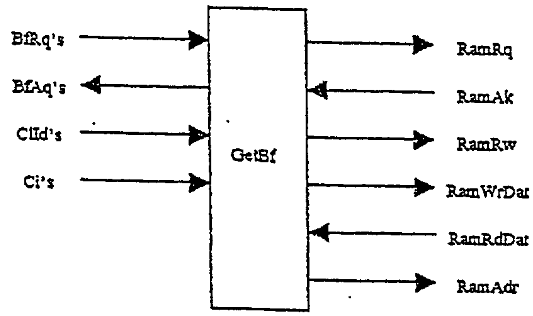

BfRq's
BfAq's
ClId's
Ci's
GetBf
RamRq
RamAk
RamRw
RamWrDat
RamRdDat
RamAdr

Fig. 41

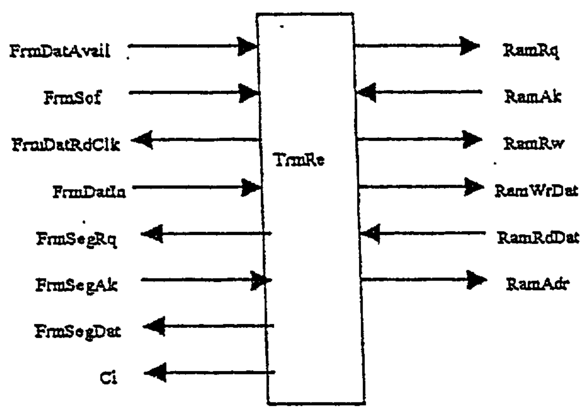

FrmDatAvail
FrmSof
FrmDatRdClk
FrmDatIn
FrmSegRq
FrmSegAk
FrmSegDat
Ci
TrmRe
RamRq
RamAk
RamRw
RamWrDat
RamRdDat
RamAdr

Fig. 42

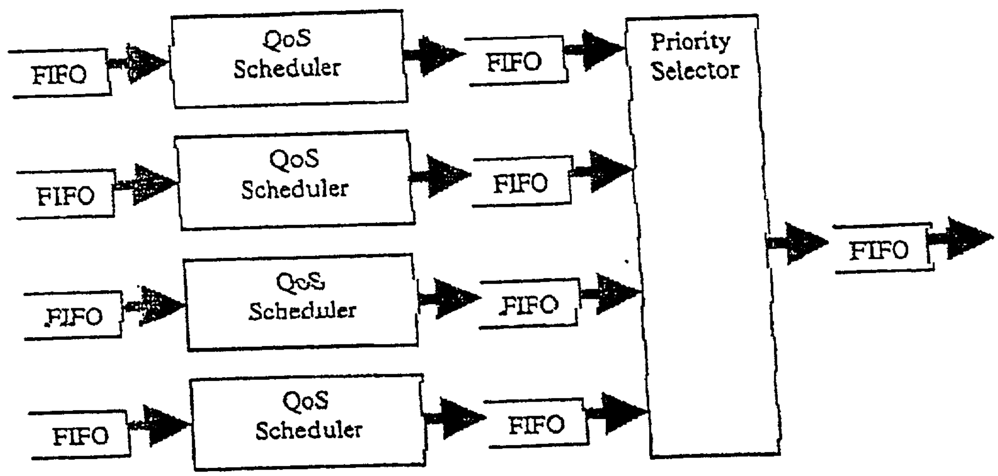

FIFO
QoS Scheduler
FIFO
Priority Selector
FIFO
QoS Scheduler
FIFO
FIFO
QoS Scheduler
FIFO
FIFO
QoS Scheduler
FIFO
FIFO

Fig. 43

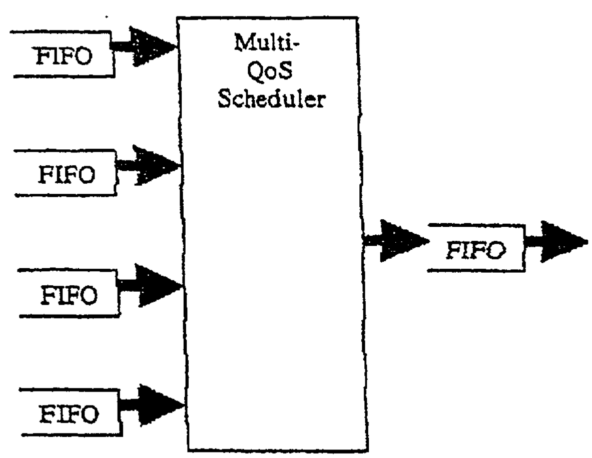

FIFO
Multi-QoS Scheduler
FIFO
FIFO
FIFO
FIFO

Fig. 44

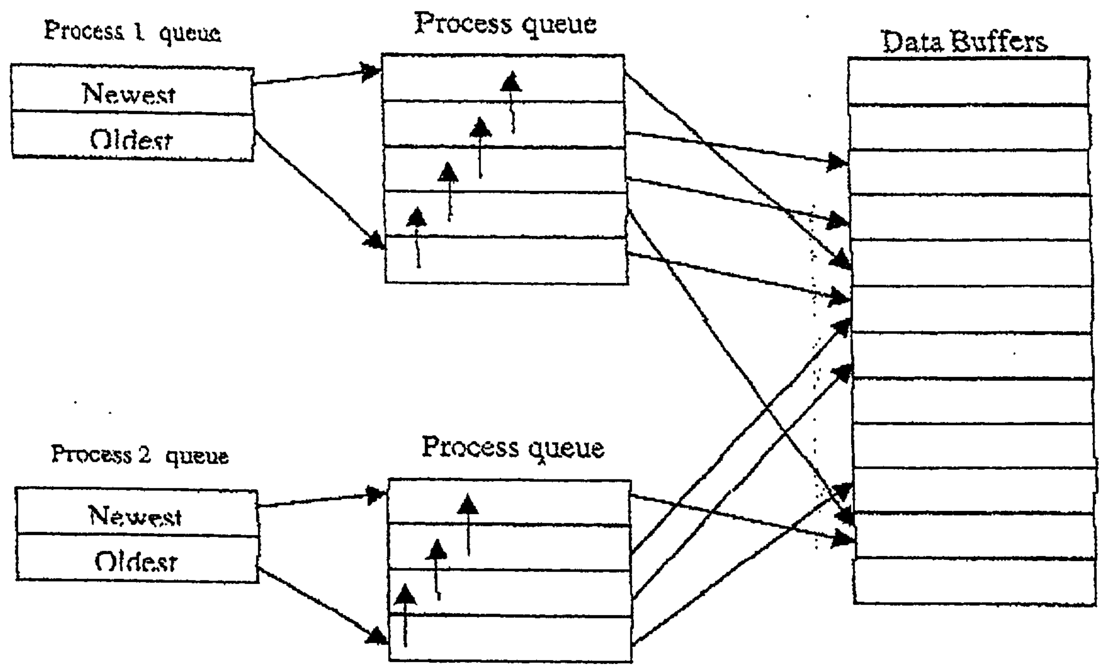

Process 1 queue
Newest
Oldest
Process queue
Data Buffers
Process 2 queue
Newest
Oldest
Process queue

Fig. 45

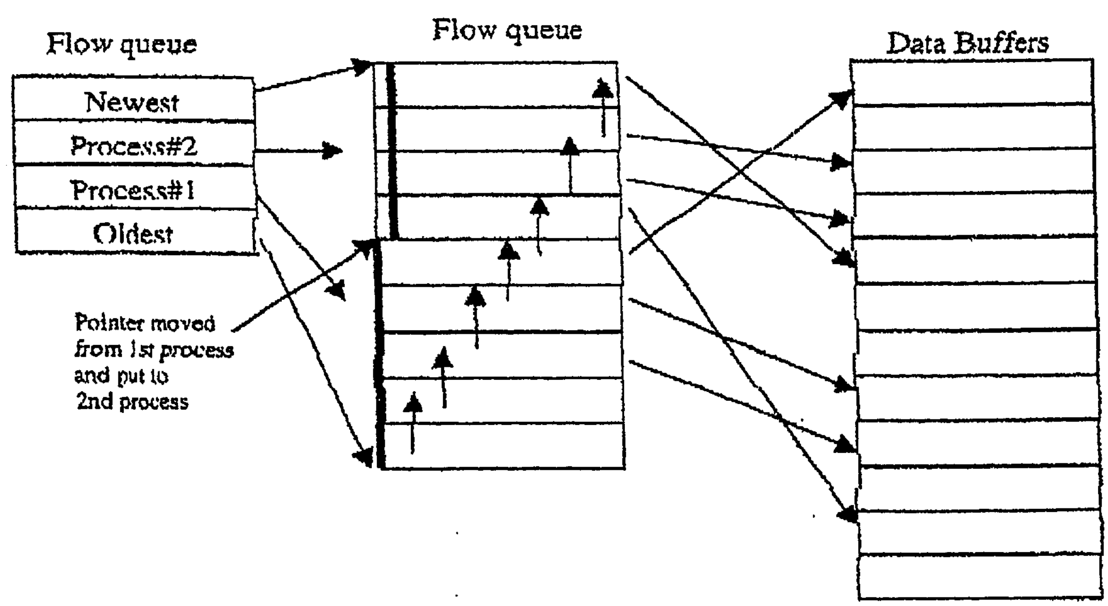

Flow queue
Newest
Process#2
Process#1
Oldest
Flow queue
Data Buffers
Pointer moved from 1st process and put to 2nd process

Fig. 46

Module	Description	Page
ATM/FR T1/E1	Network module to connect to ATM or Frame Relay WAN.	3-5
ATM T1 or E1 IMA	Network module to connect to ATM WAN. Supports grouping of multiple ATM links into single VC.	3-6
ATM DS-3/E3	Network module to connect to ATM WAN.	3-7
ATM OC-3/STM-1	Network module to connect to ATM WAN.	3-8
ATM/FR SDSL	Network module to connect to ATM or Frame Relay WAN over DSL.	3-9
ATM/FR HDSL2	Network module to connect to DSL WAN. Configurable for ATM or Frame Relay.	3-10
FR V.35/X.21	Network or User module to connect router or other Frame Relay device.	3-11
Switched 10/100BaseT	User module used to connect local Ethernet hub or switch.	3-12
PBX T1/E1/PRI	User module to connect local PBX equipment.	3-13
PBX T1/E1/PRI + BRI	User module to connect local PBX and ISDN BRI.	3-15
ISDN BRI	User module to connect up to 3 S/T/U devices.	3-16

Table 1 Module List Description

United States Patent & Trademark Office

Office of Initial Patent Examination -- Scanning Division

Application deficiencies found during scanning:

☐ Page(s)____________of__________________________________ were not present for scanning.
(Document title)

☐ Page(s)____________of__________________________________ were not present for scanning.
(Document title)

☒ ***Scanned copy is best available.*** Drawings